



САНКТ-ПЕТЕРБУРГСКИЙ
ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

ПРИОРИТЕТНЫЙ
НАЦИОНАЛЬНЫЙ ПРОЕКТ
"ОБРАЗОВАНИЕ"



Проект «Инновационная образовательная среда в классическом университете»

Пилотный проект № 22 «Разработка и внедрение
инновационной образовательной программы «Прикладные математика и физика»»

Физический факультет

Научно-образовательный центр «Электрофизика»

М.А. Павлейно, В.М. Ромаданов, Ю.Ф. Сафронова, А.А. Статуя

**Моделирование работы электрических схем в Simulink.
Применение операционных усилителей для фильтрации,
усиления и генерации сигналов.**

Учебно-методическое пособие

Санкт – Петербург
2007 г.

УДК 621.391.26 (075)

- **Рецензенты: доцент кафедры радиофизики, к.ф.-м.н. Пинегин А.Н.**
- **Печатается по решению методической комиссии физического факультета СПбГУ**
- **Рекомендовано Ученым советом физического факультета СПбГУ**

**Павлейно М.А, Ромаданов В.М., Сафронова Ю.Ф., Статуя А.А.
Моделирование работы электрических схем в Simulink.
Применение операционных усилителей для фильтрации,
усиления и генерации сигналов. – СПб., 2007**

В учебно-методическом пособии рассмотрены вопросы, связанные с работой различных электрических схем, в том числе построенных на основе операционных усилителей. Особое внимание уделяется компьютерному моделированию протекающих в них процессов. В качестве среды моделирования выбран Simulink.

Пособие предназначено для студентов 2-4-го курсов направления «Прикладные математика и физика». Оно может быть использовано при изучении дисциплины «Физические основы методов обработки и передачи информации», при выполнении курсовых и дипломных работ.

Содержание

1. Введение.....	5
2. Моделирование работы электротехнических устройств в MATLAB	6
2.1. Начало работы в Simulink	7
2.2. Примеры построения Simulink-моделей.....	13
2.2.1. Амплитудно-модулированный сигнал.....	13
2.2.2. Генерирование импульсного сигнала.....	29
2.2.3. Использование блока From Workspace	34
2.2.4. Использование передаточной функции Transfer Fcn.....	37
2.2.5. Модель системы, заданная дифференциальным уравнением	40
2.3. Установка параметров расчета и его выполнение	43
2.4. Основные блоки Simulink.....	47
2.5. Подсистемы в Simulink	59
2.6. Маскирование подсистем в Simulink	68
2.7. Использование Simulink LTI-Viewer для анализа динамических систем.	86
2.8. Начало работы в SimPowerSystems	91
2.9. Основные блоки библиотеки SimPowerSystems	97
2.10. Powergui – графический интерфейс пользователя.....	100
2.11. Примеры построения SPS-моделей.....	103
2.11.1. Зарядка конденсатора	103
2.11.2. RC-фильтр низких частот	108
2.11.3. Цепочка Вина.....	110
2.11.4. Диод.....	113
2.11.5. Создание электротехнических блоков пользователя	117
3. Свойства операционного усилителя.....	121
4. Моделирование операционного усилителя в MATLAB	124
5. Применение операционного усилителя в различных схемах.....	128
5.1. Активные фильтры	128
5.1.1. Основные сведения о фильтрах.....	129
5.1.2. Понятие обратной связи.....	130
5.1.3. Основные типы активных фильтров: схема первого типа.....	133
5.1.4. Активный RC-фильтр низких частот второго порядка.....	136
5.1.5. Активный RC-фильтр высоких частот второго порядка	146
5.1.6. Режекторный активный RC-фильтр	150
5.1.7. Основные типы активных фильтров: схема второго типа.....	158
5.1.8. Примеры активных полосопропускающих фильтров второго типа	160
5.1.9. Активные фильтры первого порядка.....	170
5.2. Усилители	186
5.2.1. Инвертирующий и неинвертирующий усилители.....	186
5.3. Генераторы сигналов на ОУ.....	191

5.3.1. Условия генерации	191
5.3.2. LC -генератор.....	192
5.3.3. Генератор с мостом Вина.....	200
5.3.4. Генератор импульсных сигналов	203
6. Список литературы.....	210

1. Введение

В данном учебно-методическом пособии рассматривается работа различных устройств, собранных на базе операционных усилителей (ОУ), основы описания электрических схем с обратными связями и способы применения ОУ в различных радиотехнических устройствах для фильтрации, генерации и усиления сигналов. Параллельно с традиционным теоретическим описанием электрических схем приводятся способы их моделирования в **MATLAB** и его пакетах расширения **Simulink** и **SimPowerSystems**. В учебно-методическом пособии не рассматривается внутреннее устройство самого ОУ и конкретные схемы его реализации на интегральных микросхемах. Вместо этого подробно разбираются физические принципы работы электрических схем, содержащих ОУ, и методы их компьютерного моделирования. Эти особенности изложения связаны с тем, что данное пособие предназначено для студентов направления «Прикладные математика и физика» физического факультета СПбГУ. Целью пособия является методическое сопровождение курса лекций «Физические основы методов обработки и передачи информации» и учебной лаборатории «Методов обработки и передачи информации». К пособию прилагаются Simulink-модели и программы MATLAB всех рассмотренных примеров.

2. Моделирование работы электротехнических устройств в MATLAB

Для лучшего понимания работы рассматриваемых нами устройств будем параллельно с описанием конкретных электрических схем проводить компьютерное моделирование их работы. Такое моделирование удобно выполнять с использованием MATLAB и его пакетов расширения **Simulink** и **SimPowerSystems**. Эти пакеты являются частью системы **MATLAB** и полностью с ней интегрируются, позволяя пользоваться преимуществами обработки и визуализации данных в **MATLAB**. В то же время эти пакеты являются достаточно самостоятельными инструментами, так что работать с ними могут и пользователи мало знакомые с командами **MATLAB**. Удобство работы в **Simulink** и **SimPowerSystems** заключается также в том, что в них реализуется принцип визуального программирования, в соответствии с которым, пользователь на экране из библиотеки стандартных блоков создает модель устройства и осуществляет расчеты. При этом, в отличие от классических способов численных расчетов, пользователю не нужно досконально изучать язык программирования и численные методы, а достаточно общих знаний, требующихся при работе на компьютере и, естественно, знаний той предметной области, в которой он работает.


Simulink предназначен для моделирования, имитации и анализа динамических систем. Он дает возможность строить графические блок-диаграммы, имитировать динамические системы, исследовать работоспособность систем и совершенствовать проекты. Такие широкие возможности позволяют **Simulink** находить применение в различных областях: физике, математике, биологии, экономике, медицине и др., везде, где задачи описываются математическими и логическими операциями. В отличие от **Simulink** пакет **SimPowerSystems** является специализированным и предназначен для имитационного моделирования электротехнических устройств. При работе с пакетами приложения **MATLAB** пользователь имеет возможность модернизировать библиотечные блоки, создавать свои собственные, а также составлять новые библиотеки блоков, возможность, которую мы будем использовать в своей работе.

Данное пособие адресовано студентам, имеющим определенные навыки работы в **MATLAB**, но, возможно, не знакомым с его пакетами расширения. Поэтому, мы опишем начало работы в **Simulink** и **SimPowerSystems**, включая графический интерфейс, и основные блоки для моделирования задач, связанных с работой устройств, собранных на базе ОУ. Студенты, знакомые с основами работы в **Simulink** и **SimPowerSystems**, могут пропустить следующую главу или просмотреть только отдельные ее части.

Все примеры будут приводиться в версии **MATLAB 7.1** и **Simulink Release 14 Service Pack 3 (R14SP3)**. Это замечание необходимо, т. к. разные версии **Simulink** могут заметно отличаться друг от друга.

2.1. Начало работы в Simulink

Для запуска **Simulink** можно сначала запустить **MATLAB** и далее выполнить одно из следующих действий:

1. для открытия библиотеки блоков **Simulink** выполнить в командной строке главного окна **MATLAB** команду **simulink** (Рис. 1) или нажать кнопку  (**Simulink**) на панели инструментов командного окна **MATLAB**.

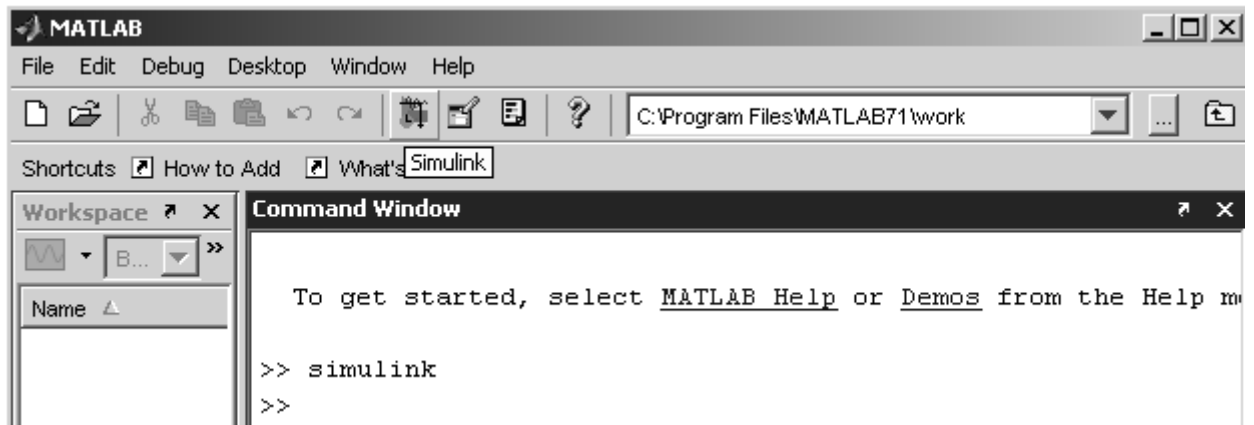


Рис. 1. Основное окно MATLAB

2. для открытия уже имеющейся **Simulink**-модели можно выполнить команду **Open...** в меню **File** и открыть файл модели (mdl-файл) или с помощью мыши перетащить нужный файл из окна проводника в окно **MATLAB**.

Использование одного из перечисленных в п.1 способов приводит к открытию окна обозревателя разделов библиотеки **Simulink** (Рис. 2).

Окно обозревателя библиотеки блоков содержит следующие элементы:

1. Заголовок с названием окна – **Simulink Library Browser**.
2. Меню с командами **File, Edit, View, Help**.
3. Панель инструментов с ярлыками наиболее часто используемых команд и строкой для быстрого поиска блоков по названию.
4. Окно комментария, поясняющего сообщения о выбранном блоке.
5. Список разделов библиотеки, реализованный в виде дерева.
6. Окно содержимого раздела библиотеки (список вложенных разделов библиотеки или блоков).
7. Строка состояния, содержащая подсказку по выполняемому действию.

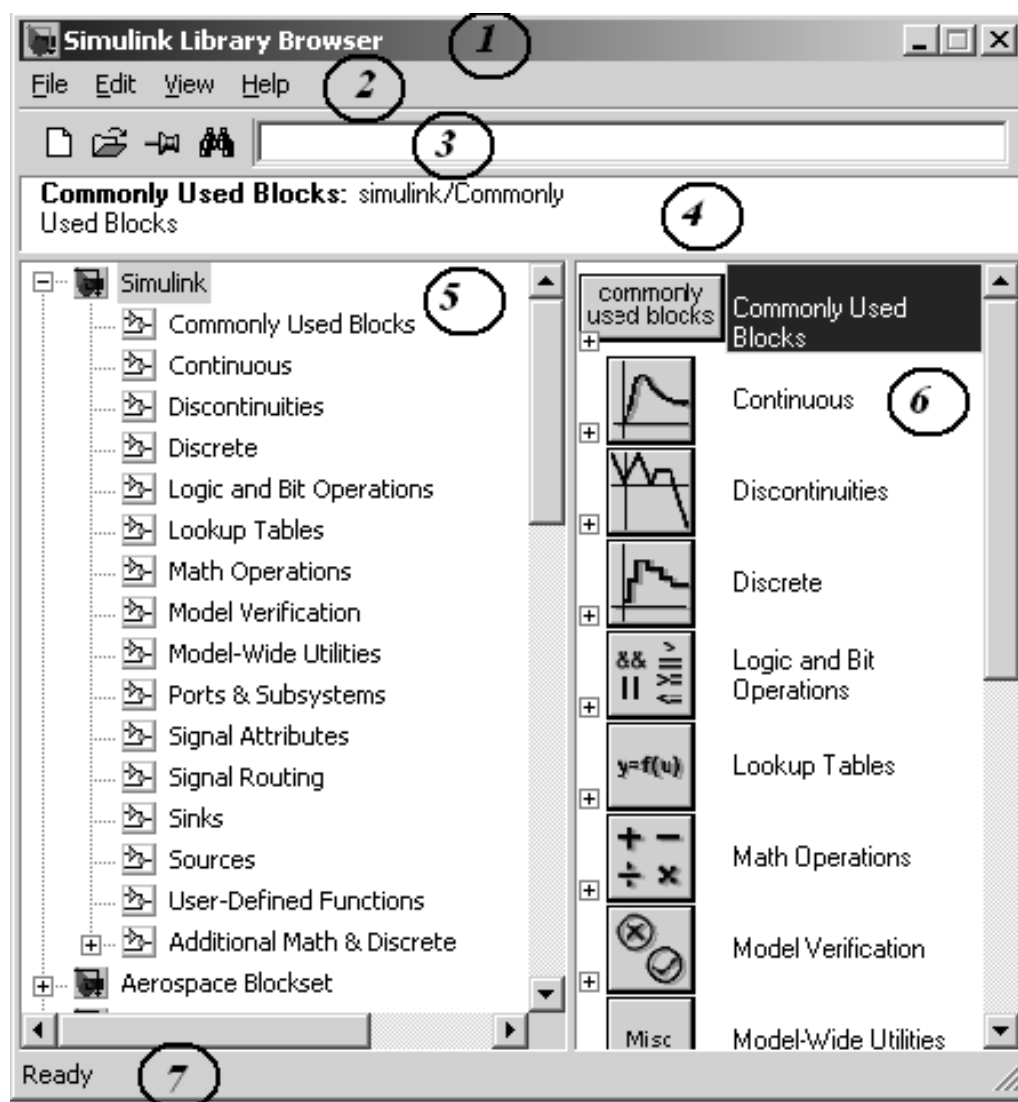


Рис. 2. Окно обозревателя разделов библиотеки Simulink

Кнопки панели инструментов имеют следующее назначение (рис.3)

1. Создать новую **Simulink**-модель (открыть новое окно модели).
2. Открыть одну из существующих **Simulink**-моделей.
3. Установить режим отображения окна обозревателя «поверх всех окон». Повторное нажатие отменяет такой режим.
4. Поиск блока по названию (по первым символам названия). После того, как блок будет найден, в окне обозревателя откроется соответствующий раздел библиотеки, а блок будет выделен. Если же блок с таким названием отсутствует, то в окне комментария будет выведено сообщение **Not found <имя блока>** (Блок не найден).

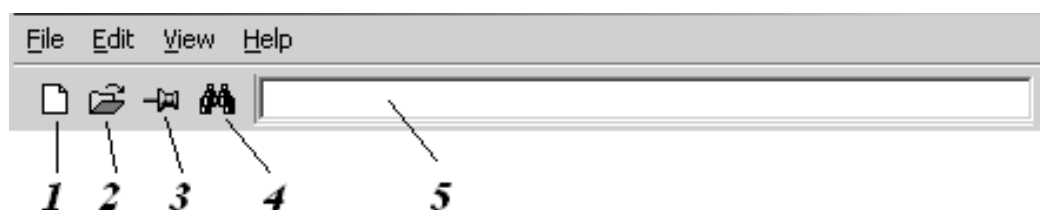



Рис. 3. Панель инструментов Simulink

Для начала работы нужно открыть рабочее окно **Simulink**, в котором в дальнейшем будет создаваться модель системы в виде структурной схемы. Это можно сделать, нажав левой кнопки мыши пиктограмму  (**Create a new model**) на панели инструментов (Рис. 3), или с помощью меню File/New/Model (Рис. 4). Также можно использовать «горячие клавиши» **Ctrl+N**.

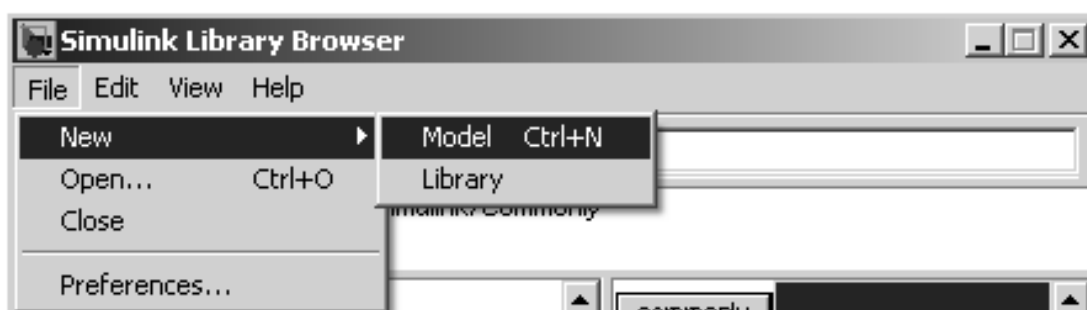


Рис. 4. Создание новой Simulink-модели

Имя появившегося окна модели (и самой **Simulink**-модели) по умолчанию задается «untitled». На (Рис. 5) изображено рабочее окно модели.

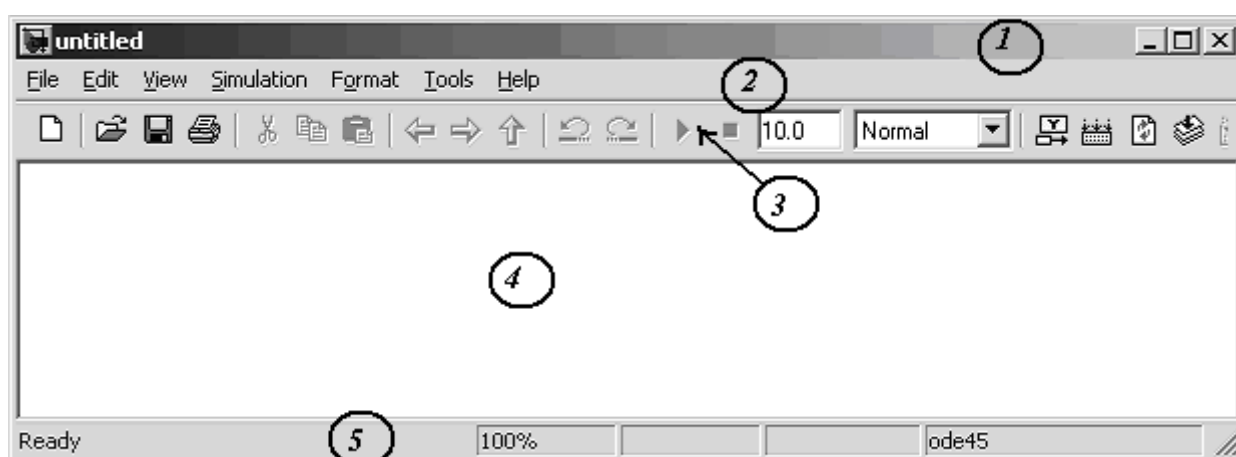


Рис. 5. Рабочее окно Simulink-модели

Для построения структурной схемы модели нужно выбрать блок из библиотеки блоков и, не отпуская левую кнопку мыши, перетащить его в рабочее окно модели. Затем следует провести линии связи, выбрать параметры

моделирования, выполнить моделирование и проанализировать результаты. Мы в дальнейшем рассмотрим подробнее эти действия на примерах.

Окно модели содержит следующие элементы:

1. Заголовок, с названием окна. Вновь созданному окну присваивается имя **Untitled** с соответствующим номером.
2. Меню с командами **File, Edit, View** и т.д.
3. Панель инструментов.
4. Окно для создания схемы модели.
5. Строка состояния, содержащая информацию о текущем состоянии модели.

Меню окна содержит команды для редактирования модели, ее настройки и управления процессом расчета, работы файлами и т.п.:

- **File (Файл)** – Работа с файлами моделей.
- **Edit (Редактирование)** – Изменение модели и поиск блоков.
- **View (Вид)** — Управление показом элементов интерфейса.
- **Simulation (Моделирование)** – Задание настроек для моделирования и управление процессом расчета.
- **Format (Форматирование)** – Изменение внешнего вида блоков и модели в целом.
- **Tools (Инструментальные средства)** – Применение специальных средств для работы с моделью (отладчик, линейный анализ и т.п.).
- **Help (Справка)** – Вывод окон справочной системы.

Некоторые из этих команд очевидны, а некоторые мы подробнее рассмотрим в дальнейшем.

Для работы с моделью можно также использовать кнопки на панели инструментов (Рис. 6).

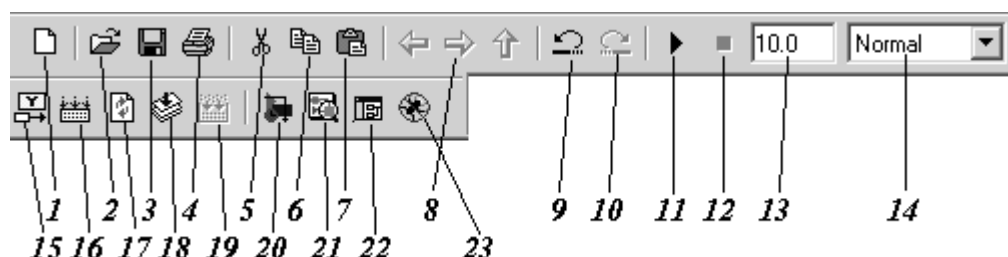



Рис. 6. Панель инструментов рабочего окна модели

Кнопки панели инструментов имеют следующее назначение:

1. **New Model** – Открыть новое (пустое) окно модели.

2. **Open Model** – Открыть существующий **mdl**-файл.
3. **Save Model** – Сохранить **mdl**-файл.
4. **Print Model** – Вывод на печать блок-диаграммы модели.
5. **Cut** – Вырезать выделенную часть модели в буфер промежуточного хранения.
6. **Copy** – Скопировать выделенную часть модели в буфер промежуточного хранения.
7. **Paste** – Вставить в окно модели содержимое буфера промежуточного хранения.
8. **Navigate** – Кнопки навигации между окнами **Navigate back**, **Navigate forward**. **Go to parent system** — Переход из подсистемы в систему высшего уровня иерархии (“родительскую систему”). Команда доступна только тогда, когда открыта подсистема.
9. **Undo** – Отменить предыдущую операцию редактирования.
10. **Redo** – Восстановить результат отмененной операции редактирования.
11. **Start/Pause/Continue Simulation** – Запуск модели на исполнение (команда **Start**); после запуска модели на изображении кнопки выводится символ , и ей соответствует уже команда **Pause** (Приостановить моделирование). Для возобновления моделирования следует щелкнуть по той же кнопке, поскольку в режиме паузы ей соответствует команда **Continue** (Продолжить).
12. **Stop** – Закончить моделирование. Кнопка становится доступной после начала моделирования, а также после выполнения команды **Pause**.
13. **Simulation stop time** – Время окончания моделирования.
14. **Normal/Accelerator** – Обычный/Ускоренный режим расчета. Инструмент доступен, если установлено приложение **Simulink Performance Tool**.
15. **Show when hovering** – Показывать при «зависании».
16. **Incremental build** – Компилирует модель, используя **Run-Time Workshop**.
17. **Refresh model blocks** – Обновляет блоки модели, вызывая соответствующие процессы, чтобы отобразить изменения интерфейса.
18. **Update diagram** – Обновляет диаграмму (изображение схемы модели), при этом происходит повторная инициализация всех блоков и подсистем.
19. **Build subsystem** – Компилирует выделенную подсистему в исполняемый файл (*.exe). Скомпилированная программа выполняет те же действия, что и подсистема, но выполняется независимо от **Simulink**. В результате выполнения программы создаётся **mat**-файл с выходными данными подсистемы.
20. **Library Browser** – Открыть окно обозревателя библиотек.
21. **Launch model explorer** – Запускает утилиту **Model Explorer**, которая предоставляет удобный графический интерфейс, в котором объедине-

ны все настройки **Simulink**-модели, подсистем и отдельных блоков, рабочие области **MATLAB**, модели и подсистем.

22. Toggle Model Browser – Открыть окно обозревателя модели.

23. Debug — Запуск отладчика модели.

В нижней части окна модели находится строка состояния, в которой отображаются краткие комментарии к кнопкам панели инструментов, а также к пунктам меню, когда указатель мыши находится над соответствующим элементом интерфейса. Это же текстовое поле используется и для индикации состояния **Simulink**: **Ready** (Готов) или **Running** (Выполнение). В строке состояния отображаются также:

- масштаб отображения блок-диаграммы (в процентах, исходное значение равно 100%),
- индикатор степени завершенности сеанса моделирования (появляется после запуска модели),
- текущее значения модельного времени (выводится также только после запуска модели),
- используемый алгоритм расчета состояний модели (метод решения).

Готовые блоки, которые можно использовать для построения структурных схем, содержатся в библиотеке **Simulink** (пункт 5 на Рис. 2). Список разделов библиотеки **Simulink** представлен в виде дерева, и правила работы с ним являются общими для списков такого вида:

- Пиктограмма свернутого узла дерева содержит символ "+", а пиктограмма развернутого содержит символ "-".
- Для того чтобы развернуть или свернуть узел дерева, достаточно щелкнуть на его пиктограмме левой клавишей мыши.

При выборе соответствующего раздела библиотеки в правой части окна отображается его содержимое. Библиотека **Simulink** довольно обширна и в ней модели объектов и устройств (блоки) группируются по функциональному уровню в разделы. Мы рассмотрим только те разделы и те блоки, которые будут использоваться нами в дальнейшем:

1. **Continuous** – блоки для описания линейных систем (шаг по времени определяется автоматически).
2. **Discrete** – для описания линейных систем с фиксированным шагом по времени.
3. **Math Operations**– блоки математических операций.
4. **Nonlinear** – нелинейные блоки.
5. **Ports & Subsystems** – порты и подсистемы.
6. **Signal Routing** – маршрутизация сигналов.
7. **Sinks** – регистрирующие устройства.

8. **Sources** – источники сигналов и воздействий.

9. **User-Defined Function** – функции, заданные пользователем.

Большинство **Simulink**-моделей укладываются в следующую общую схему (Рис. 7).

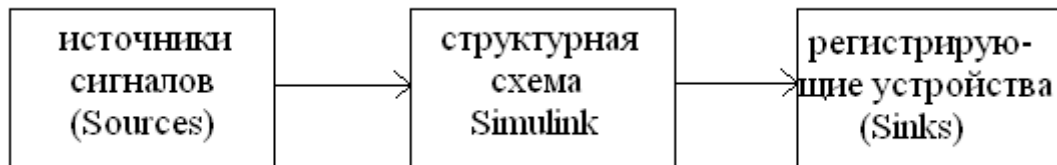


Рис. 7. Общая схема **Simulink**-моделей

Выходной сигнал источника является переменной, которая воздействует на систему, описываемую в виде структурной схемы. Преобразованный сигнал источника, после прохождения через структурную схему, регистрируется с помощью специальных устройств. **Simulink** -модели могут включать один или несколько блоков источников и регистраторов. Но приведенная выше схема не является универсальной, т. к. модели могут не содержать блоки из каких-нибудь выделенных на схеме групп как, например, простые модели, состоящие только из источников сигналов и регистрирующих устройств, или более сложные модели автоколебательных систем. На структурной схеме такие системы не имеют входного воздействия.

2.2. Примеры построения **Simulink**-моделей

2.2.1. Амплитудно-модулированный сигнал

В качестве примера рассмотрим построение амплитудно-модулированного сигнала (Рис. 8). **Simulink**-модель с этим примером называется **ampl_modul.mdl**. Зависимость сигнала от времени t определяется выражением $A(t) \sin \omega_0 t$, где амплитуда A изменяется по следующему закону: $A(t) = A_{MAX} (1 + m \sin \Omega t)$ с частотой $\Omega \ll \omega_0$. Для характеристики относительного изменения амплитуды модулированных колебаний используют параметр $m = \frac{(A_{MAX} - A_{MIN})}{(A_{MAX} + A_{MIN})}$ – коэффициент модуляции.

Запишем сигнал следующим образом:

$$x(t) = A_{MAX} (1 + m \sin \Omega t) \sin \omega_0 t = (A_{MAX} + A_{MAX} m \sin \Omega t) \sin \omega_0 t \quad (1)$$

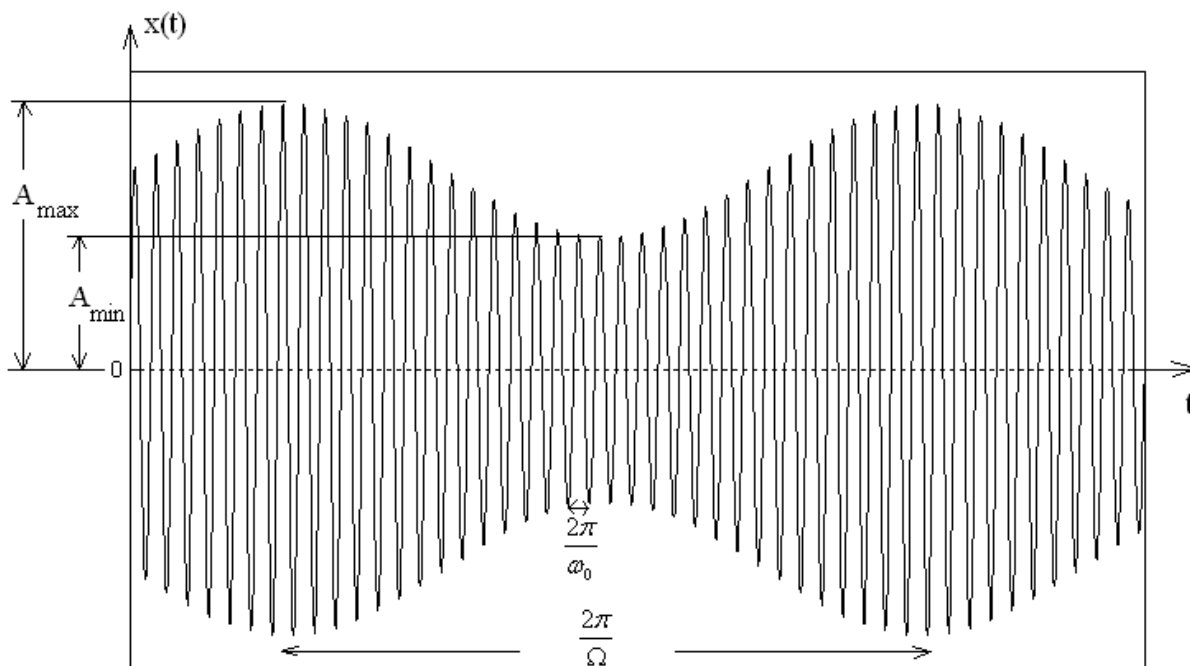


Рис. 8. График амплитудно-модулированной функции

Это выражение мы и будем «собирать» из блоков в **Simulink**. Зададим для определенности следующие параметры сигнала:

$$\begin{aligned} A_{max} &= 10; \\ \omega_0 &= 2\pi \cdot 5 \text{ рад/с}; \\ \Omega &= 2\pi \cdot 0,1 \text{ рад/с}; \\ m &= 5/15. \end{aligned}$$

Для построения модели из раздела «Источники сигналов и воздействий» (**Sources**) библиотеки **Simulink** перетащим в рабочее окно модели блок синусоидального сигнала. Его пиктограмма изображена на Рис. 9.

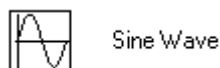



Рис. 9. Пиктограмма блока синусоидального сигнала

Нам потребуются два источника синусоидальных сигналов для задания частоты заполнения и частоты изменения амплитуды. Можно два раза перетащить этот блок в рабочее окно или скопировать и вставить уже имеющийся там блок синусоидального сигнала из буфера (Рис. 10). Для вставки объекта из буфера необходимо предварительно указать место вставки, щелкнув левой клавишей мыши в предполагаемом месте вставки, а затем выполнить коман-

ду **Edit/Paste** или воспользоваться инструментом  на панели инструментов.

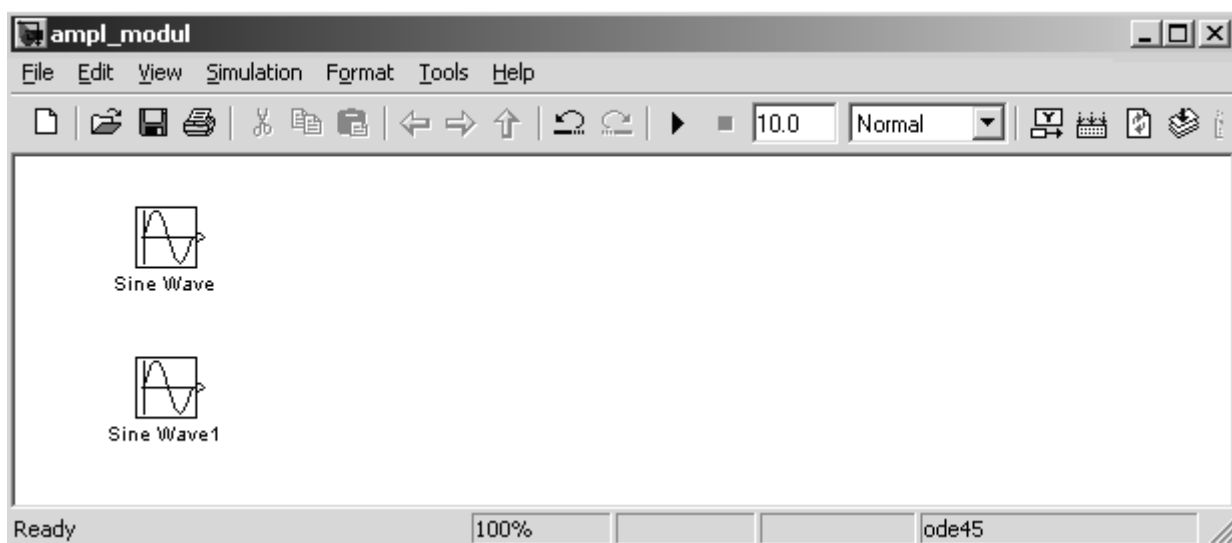


Рис. 10. Начало построения модели амплитудно-модулированного сигнала

Двойной щелчок мыши на иконке блока открывает окно, содержащее описание и параметры данного блока. Для источника гармонического сигнала это окно изображено на Рис. 11. В верхней части окна приведено общее выражение для вычисления выходного сигнала (**output signal**) $O(t)$ этого блока

$$O(t) = Amp * Sin(Freq * t + Phase) + Bias \quad (2)$$

и пояснения, касающиеся использования данного блока как источника дискретных сигналов. Из выражения (2) видно, что выходным сигналом является синусоида с амплитудой Amp , циклической частотой $Freq$, фазой $Phase$ и сдвигом по оси ординат $Bias$.

По ходу того, как у нас в примерах будут появляться новые блоки, мы будем описывать особенности их использования и возможности этих блоков. Рассмотреть все блоки **Simulink** и **SimPowerSystems** в пределах данного учебно-методического пособия не представляется возможным, поэтому ниже в таблицах будут перечислены только некоторые блоки и приведено их основное назначение.

Рассмотрим подробно окно параметров блока синусоидального сигнала (Рис. 11). В основной части окна можно задать параметры блока. Этот блок может использоваться как источник непрерывных во времени, так и дискретных сигналов.

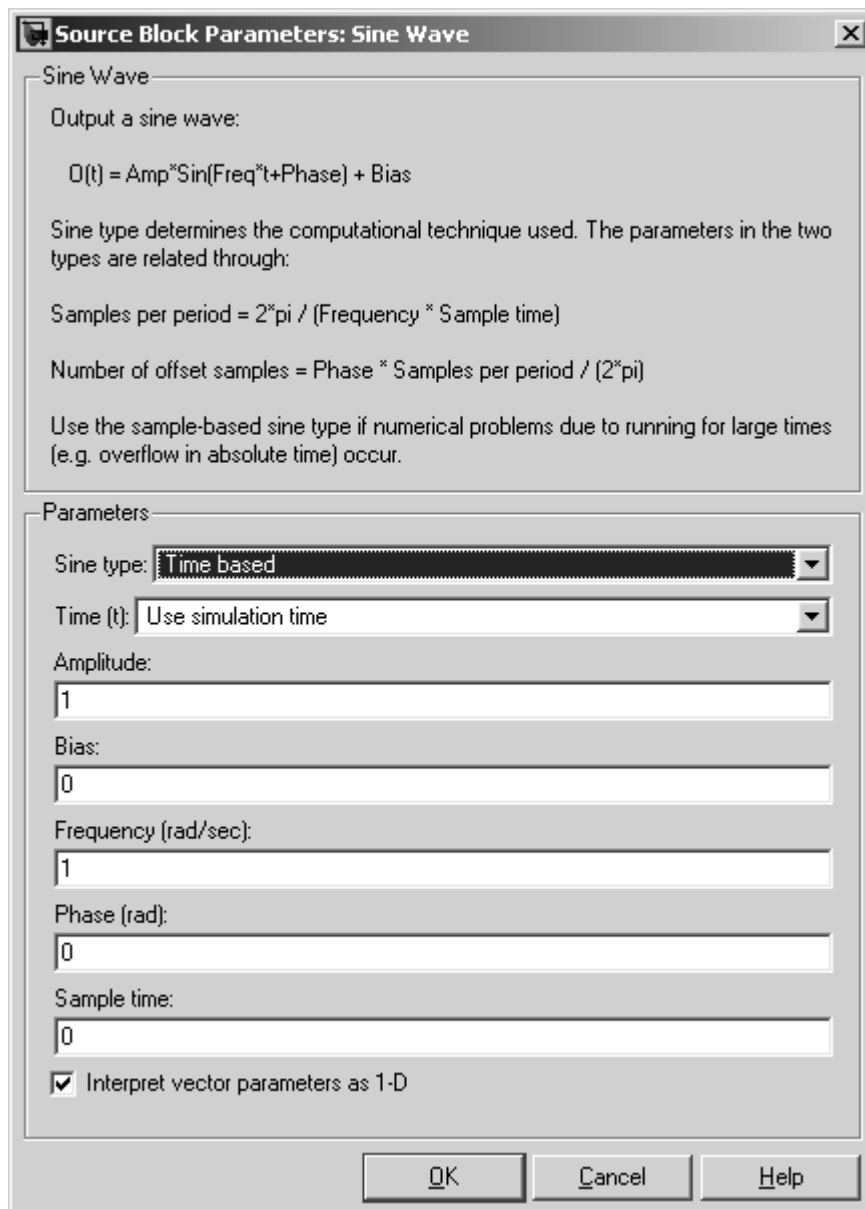


Рис. 11. Диалоговое окно источника гармонического сигнала

Выбрать тип блока можно в выпадающем списке **Sine type** (Рис. 12):

- источник непрерывного сигнала – **time based**,
- источник дискретного сигнала – **sample based**.

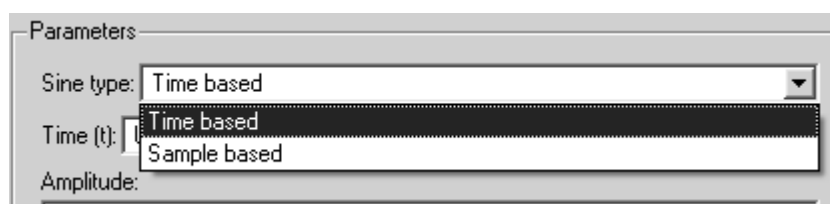


Рис. 12. Выбор источника непрерывного или дискретного сигнала

С помощью выпадающего списка **time(t)** (Рис. 13) можно определить, будет ли переменная t являться временем моделирования или задаваться внешним источником:

- используется время моделирования – **use simulation time**,
- используется внешний источник – **use external signal**.

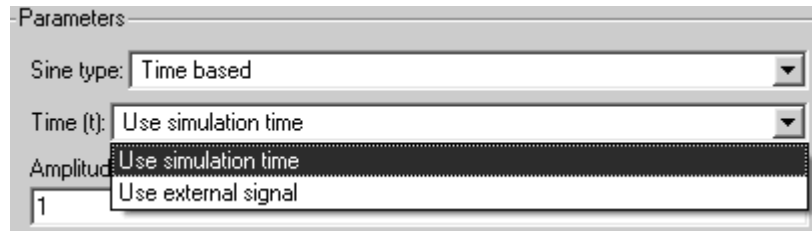


Рис. 13. Выбор типа переменной времени t

При выборе внешнего источника времени изменяются как диалоговое окно блока (остаются только соответствующие параметры), так и иконка блока (появляется входной порт, который нужно будет соединить с источником времени t). Измененная пиктограмма блока изображена на Рис. 14.

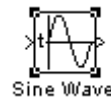


Рис. 14. Пиктограмма блока синусоидального сигнала с внешним источником времени

Мы выберем источник непрерывного сигнала с переменной t , соответствующей времени моделирования. В следующих ячейках окна можно задать параметры самого сигнала: амплитуду (**amplitude**), сдвиг по оси ординат (**bias**), частоту в рад/с (**frequency**), фазу в рад. (**phase**). Пусть максимальная амплитуда нашего модулированного сигнала будет равна 10, частота заполнения – 0,5 Гц и частота огибающей – 0,1 Гц. Тогда параметры блоков будут выглядеть, как показано на Рис. 15.

При задании численных параметров следует иметь в виду, что в качестве десятичного разделителя должна использоваться точка, а не запятая.

В последней ячейке диалогового окна параметров блока можно задать период квантования **Sample time**, если блок используется как дискретный источник сигнала, или оставить значение 0 для источника непрерывного сигнала.

Внизу окна находятся кнопки **OK** и **Apply**, подтверждающие изменения, **Cancel**, отменяющая изменения, и **Help**, открывающая окно справки по этому блоку. После внесения изменений нужно закрыть окно кнопкой **OK**.

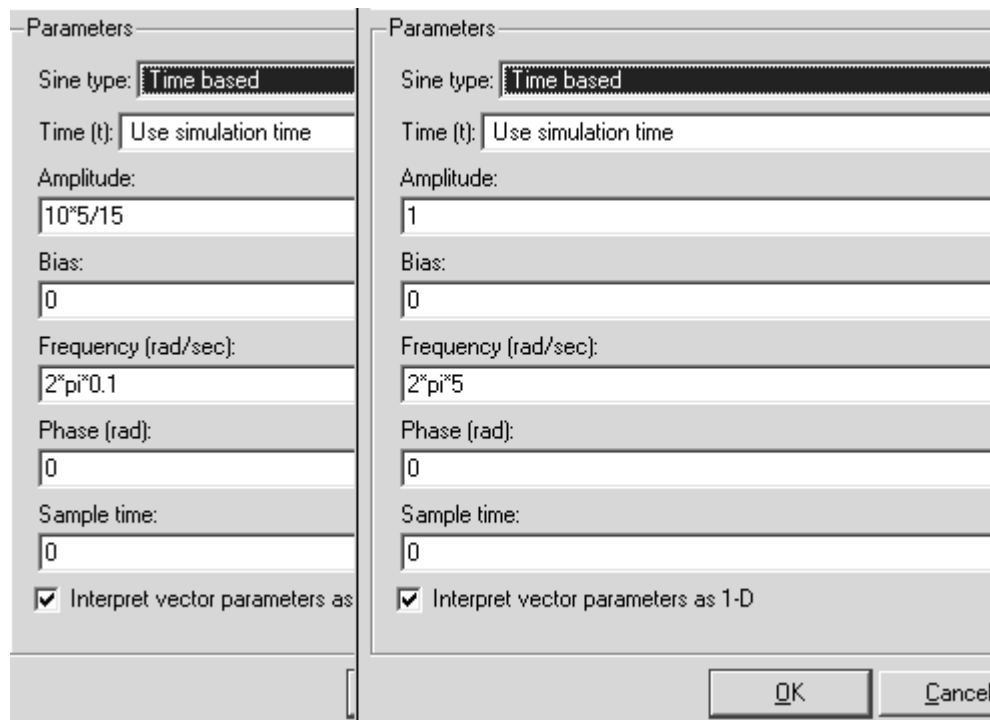


Рис. 15. Выбор параметров для блоков гармонических сигналов

Также нам потребуется еще один источник сигналов – это постоянный сдвиг $A_{max} = 10$ для амплитуды сигнала. Пиктограмма этого блока изображена на Рис. 16.



Рис. 16. Пиктограмма блока постоянного сигнала

В окне параметров этого блока (Рис. 17) задается значение константы 10.

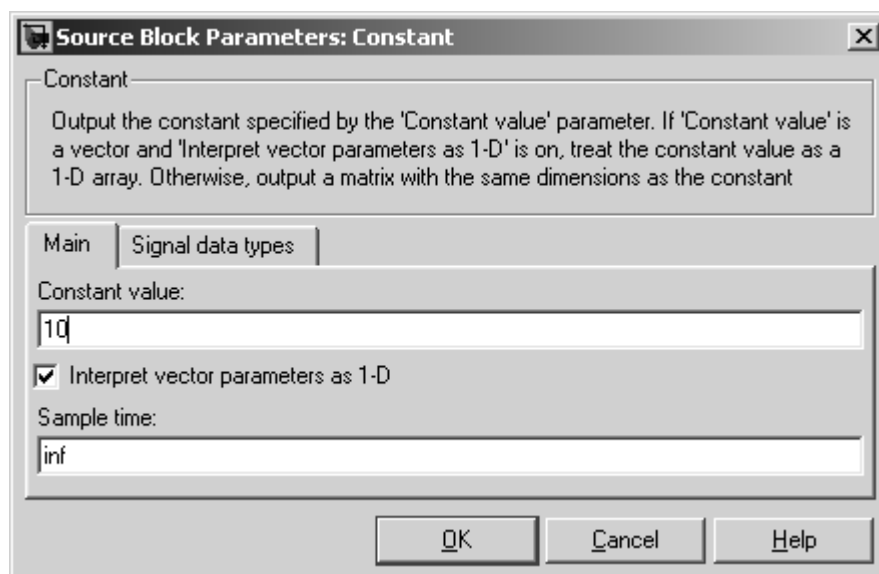


Рис. 17. Окно параметров блока постоянного сигнала

Для сложения сигналов перетаскиваем в рабочее окно из раздела «Блоки математических операций» (**Math Operations**) блок суммирования. Пиктограмма этого блока изображена на Рис. 18.



Рис. 18. Пиктограмма блока суммирование

Окно его параметров изображено на Рис. 19. Этот блок выполняет вычисление суммы текущих значений сигналов. В выпадающем списке **Icon shape** (форма блока) можно выбрать:

- **round**—окружность,
- **rectangular** – прямоугольник.

Второй параметр – **List of sign** (список знаков) определяет количество входов и операцию (сложение или вычитание). В списке можно использовать следующие знаки: + (плюс), - (минус) и | (разделитель знаков).

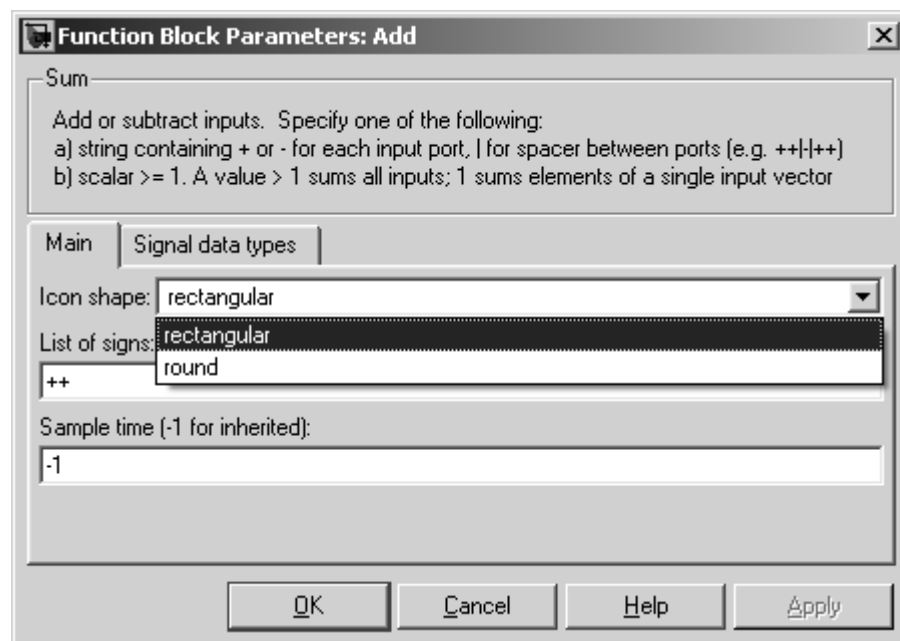


Рис. 19. Окно параметров блока суммирование

В параметре **List of sign** можно также указать число входов блока. В этом случае все входы будут суммирующими. Если количество входов блока превышает 3, то удобнее использовать блок **Sum** прямоугольной формы. Блок может использоваться для суммирования скалярных, векторных или матричных сигналов. Типы суммируемых сигналов должны совпадать. Нельзя, например, подать на один и тот же суммирующий блок сигналы целого и действительного типов. Если количество входов блока больше, чем один, то

блок выполняет поэлементные операции над векторными и матричными сигналами. При этом количество элементов в матрице или векторе должно быть одинаковым. Если в качестве списка знаков указать цифру **1** (один вход), то блок можно использовать для определения суммы элементов вектора.

Для перемножения сигналов перетаскиваем в рабочее окно также из раздела «Блоки математических операций» (**Math Operations**) блок произведения. Его пиктограмма изображена на Рис. 20.



Рис. 20. Пиктограмма блока произведения

Окно параметров этого блока изображено на Рис. 21. В этом окне можно задать число входных портов блока (число перемножаемых сигналов) и метод выполнения математической операции (поэлементное перемножение (соответствует команде **.*** MATLAB) или перемножение матриц).

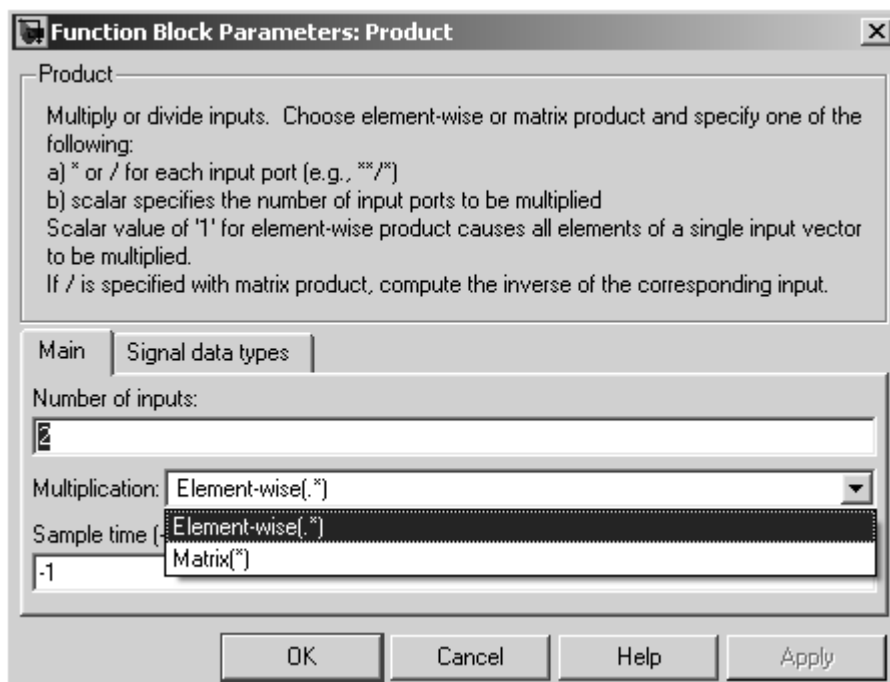


Рис. 21. Окно параметров блока умножения

Заметим, что параметр **Number of inputs** (количество входов) может задаваться как числом, так и списком знаков. В списке знаков можно использовать знаки ***** (умножить) и **/** (разделить). Если параметр **Number of inputs** задан списком, включающим, кроме знаков умножения, также знаки деления, то метки входов будут обозначены символами соответствующих операций.

Блок может использоваться для операций умножения или деления скалярных, векторных или матричных сигналов. Типы входных сигналов блока должны совпадать. Если в качестве количества входов указать цифру **1** (один

вход), то блок можно использовать для определения произведения элементов вектора.

Для визуализации сигнала будем использовать блок Осциллограф (**Scope**) из раздела «Регистрирующие устройства» (**Sinks**). С помощью этого блока можно построить графики исследуемых сигналов как функции времени, наблюдать за изменениями сигналов в процессе моделирования. Пиктограмма этого блока изображена на Рис. 22.

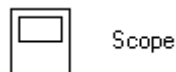


Рис. 22. Пиктограмма блока осциллограф

При двойном щелчке по этому блоку, в отличие от ранее рассмотренных блоков, появится не окно с параметрами блока, а окно для просмотра графиков (Рис. 23). Это окно можно открыть на любом этапе расчета (как до начала расчета, так и после него, а также во время расчета). В том случае, если на вход блока поступает векторный сигнал, то кривая для каждого элемента вектора строится отдельным цветом.

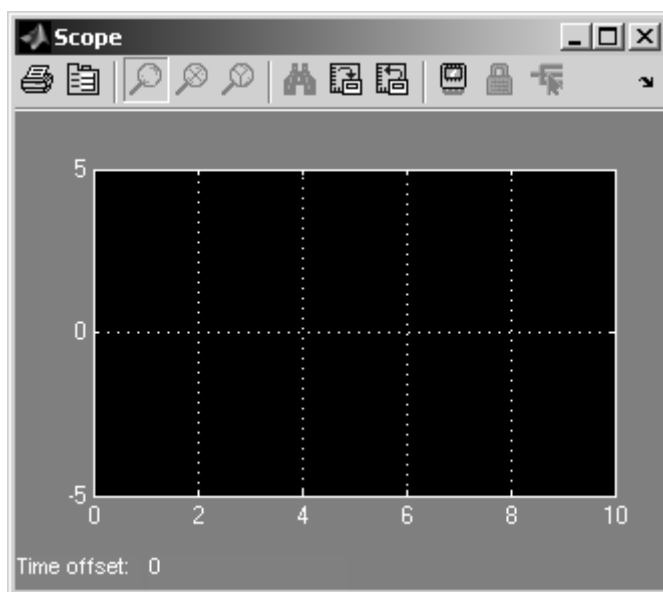


Рис. 23. Окно для просмотра графиков

Настройка окна осциллографа выполняется с помощью панелей инструментов (Рис. 24).

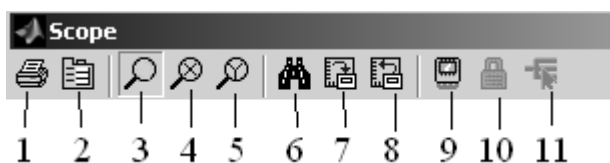


Рис. 24. Панель инструментов окна осциллографа

Панель инструментов содержит 11 кнопок:

1. **Print** – печать содержимого окна осциллографа.
2. **Parameters** – доступ к окну настройки параметров.
3. **Zoom** – увеличение масштаба по обеим осям.
4. **Zoom X-axis** – увеличение масштаба по горизонтальной оси.
5. **Zoom Y-axis** – увеличение масштаба по вертикальной оси.
6. **Autoscale** – автоматическая установка масштабов по обеим осям.
7. **Save current axes settings** – сохранение текущих настроек окна.
8. **Restore saved axes settings** – установка ранее сохраненных настроек окна.
9. **Floating scope** – перевод осциллографа в “плавающий” режим (так называется режим, когда осциллограф не соединен с остальной схемой).
10. **Lock/Unlock axes selection** – зафиксировать/разорвать связь между текущей координатной системой окна и отображаемым сигналом. Инструмент доступен, если включен режим **Floating scope**.
11. **Signal selection** – выбор сигналов для отображения. Инструмент доступен, если включен режим **Floating scope**.

Изменение масштабов отображаемых графиков можно выполнить несколькими способами:

1. Нажать соответствующую кнопку (🔍, 🔍 или 🔍) и щелкнуть один раз левой клавишей мыши в нужном месте графика. Произойдет 2,5 кратное увеличение масштаба.

2. Нажать соответствующую кнопку (🔍, 🔍 или 🔍) и, нажав левую клавишу мыши, с помощью динамической рамки или отрезка указать область графика для увеличенного изображения.

3. Щелкнуть правой клавишей мыши в окне графиков и, выбрать команду **Axes properties...** в контекстном меню. Откроется окно свойств графика, в котором с помощью параметров **Y-min** и **Y-max** можно указать предельные значения вертикальной оси. В этом же окне можно указать заголовок графика (**Title**), заменив выражение **%<SignalLabel>** в строке ввода. Окно свойств показано на Рис. 25.

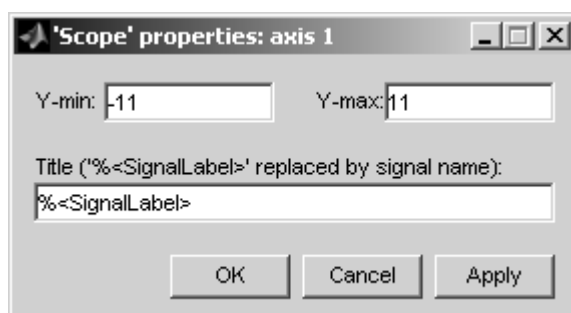



Рис. 25. Окно изменения масштаба осей осциллографа

Часто требуется изменить число графиков, выводимых в окне осциллографа, записать результаты в рабочую область **MATLAB**. Это можно сделать с помощью параметров блока осциллограф, которые устанавливаются в окне **Scope parameters**. Это окно открывается с помощью инструмента  (**Parameters**) панели инструментов. Окно параметров имеет две вкладки:

General – общие параметры,

Data history – параметры сохранения сигналов в рабочей области **MATLAB**.

Вкладки параметров осциллографа показаны на Рис. 26.

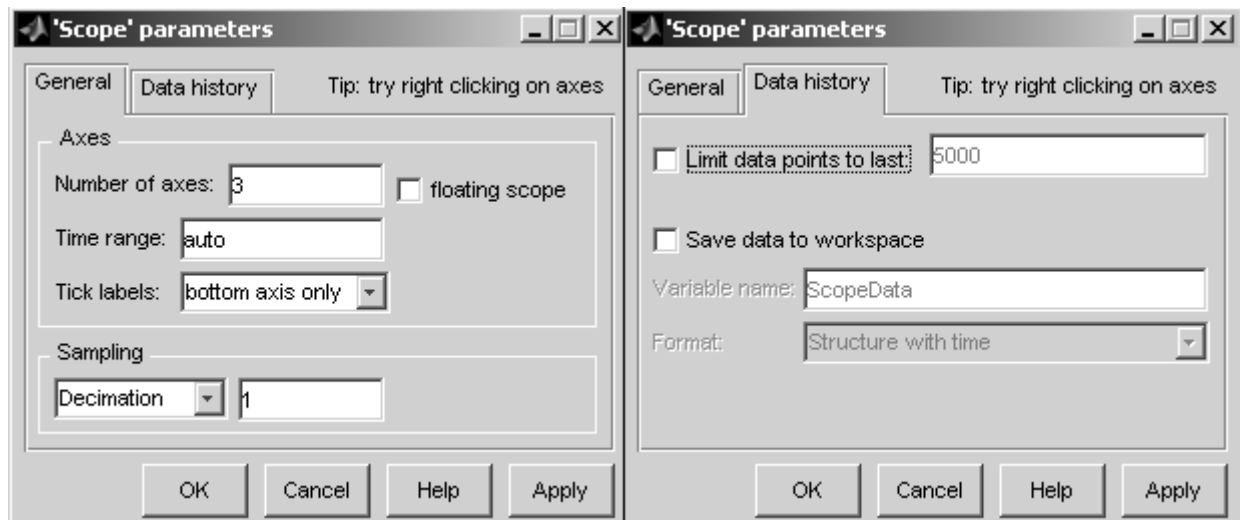


Рис. 26. Параметры осциллографа

На вкладке **General** задаются следующие параметры:

1. **Number of axes** — число входов (систем координат) осциллографа. При изменении этого параметра на изображении блока появляются дополнительные входные порты и на экране осциллографа дополнительные оси координат.

2. **Time range** — величина временного интервала, для которого отображаются графики. Если время расчета модели превышает заданное параметром **Time range**, то вывод графика производится порциями, при этом интервал отображения каждой порции графика равен заданному значению **Time range**.

3. **Tick labels** — вывод/скрытие осей и меток осей. Может принимать три значения (выбираются из списка):

4.

- **all** – подписи для всех осей,
- **none** – отсутствие всех осей и подписей к ним,
- **bottom axis only** – подписи горизонтальной оси только для нижнего графика.

5. **Sampling** — установка параметров вывода графиков в окне. Задаёт режим вывода расчетных точек на экран. При выборе **Decimation** кратность вывода устанавливается числом, задающим шаг выводимых расчетных точек. По умолчанию на экран выводится каждая расчетная точка (параметр **Decimation** равен 1), но этот порядок можно изменить. В том случае, если режим вывода расчетных точек задается как **Sample time**, то его числовое значение определяет интервал квантования при отображении сигнала.

6. **floating scope** – перевод осциллографа в “свободный” режим (при установленном флажке).

На вкладке **Data history** (Рис. 26) задаются следующие параметры:

1. **Limit data points to last** – максимальное количество отображаемых расчетных точек графика. При превышении этого числа начальная часть графика обрезается. В том случае, если флажок параметра **Limit data points to last** не установлен, то **Simulink** автоматически увеличит значение этого параметра для отображения всех расчетных точек.

2. **Save data to workspace** – сохранение значений сигналов в рабочей области **MATLAB**.

3. **Variable name** – имя переменной для сохранения сигналов в рабочей области **MATLAB**.

4. **Format** – формат данных при сохранении в рабочей области **MATLAB**. Может принимать значения:

- **Array** – массив,
- **Structure** – структура,
- **Structure with time** – структура с дополнительным полем «время».

Как мы видели, с помощью блока осциллограф можно сохранять данные в рабочей области **MATLAB**. Рассмотрим еще один блок, выполняющий эту операцию. Пиктограмма этого блока изображена на Рис. 27.

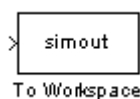


Рис. 27. Пиктограмма блока сохранения данных в рабочей области MATLAB

Этот блок находится в библиотеке **Sinks** – приемники сигналов и имеет только один вход. Окно параметров этого блока изображено на Рис. 28.

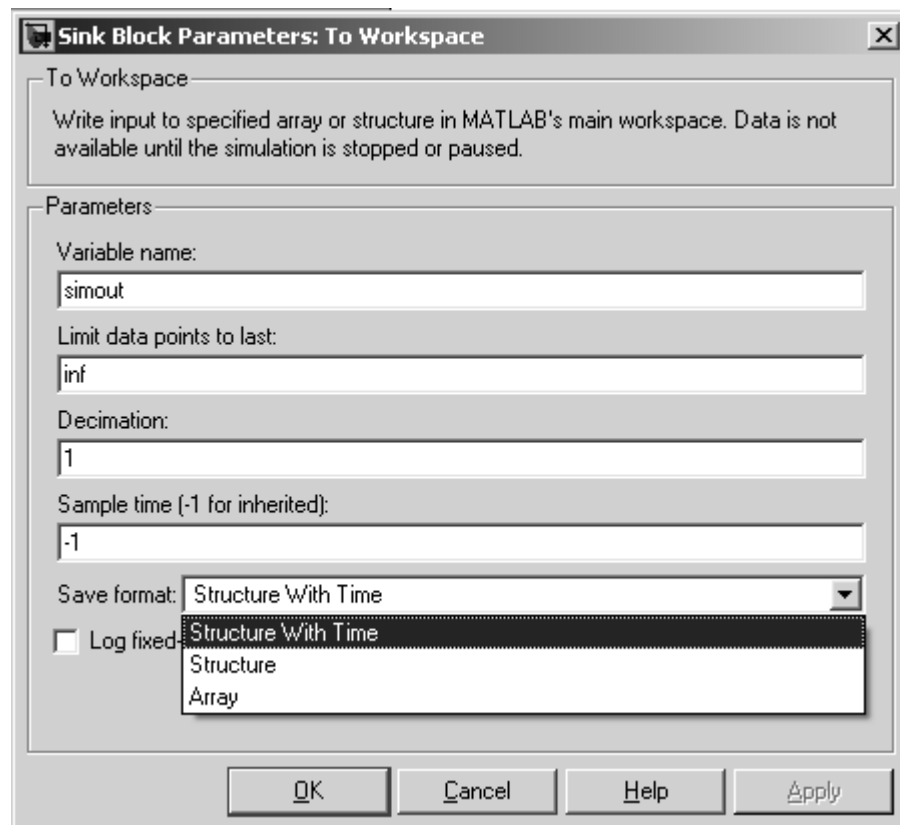


Рис. 28. Окно параметров блока сохранения данных в рабочей области MATLAB

Этот блок имеет следующие параметры:

1. **Variable name** – имя переменной, содержащей записываемые данные, по умолчанию **simout**.
2. **Limit data points to last** – максимальное количество сохраняемых расчетных точек по времени (отсчет ведется от момента завершения моделирования). В том случае, если значение параметра **Limit data points to last** задано как **inf**, то в рабочей области будут сохранены все данные.
3. **Decimation** – кратность записи данных в рабочую область. Если задана **1**, то будут записываться все отсчеты.
4. **Sample time** – шаг модельного времени. Определяет дискретность записи данных. **-1** задается для непрерывной модели.
5. **Save format** – формат сохранения данных. Может принимать значения:
 - **Array** – матрица. Данные сохраняются как массив, в котором число строк определяется числом расчетных точек по времени, а число столбцов – размерностью вектора подаваемого на вход блока. Если на вход подается скалярный сигнал, то матрица будет содержать лишь один столбец.
 - **Structure** – структура. Данные сохраняются в виде структуры, имеющей три поля: **time** – время, **signals** – сохраняемые значения

сигналов и размерность, **blockName** – имя модели и блока **To Workspace**. Поле **time** для данного формата остается не заполненным.

- **Structure with Time** – структура с дополнительным полем (временем). Для данного формата, в отличие от предыдущего, поле **time** заполняется значениями времени.

Выберем формат сохранения данных – структура со временем (**Structure with Time**).

После того, как мы перетащили в рабочее окно **Simulink** все необходимые блоки, набор блоков структурной схемы будет выглядеть следующим образом (Рис. 29).

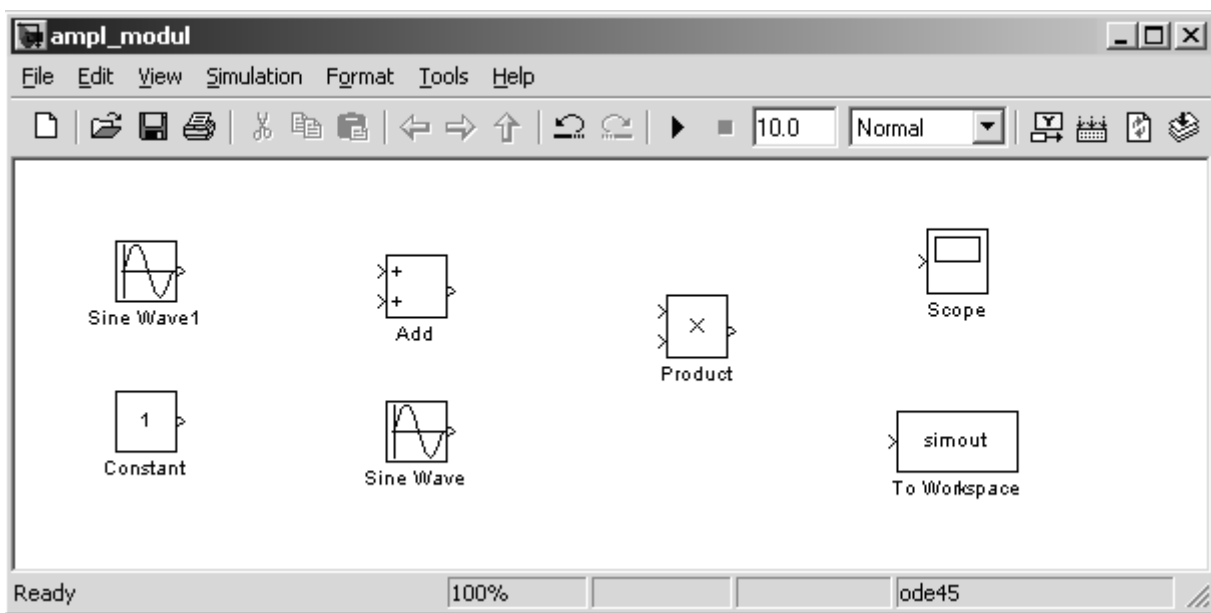


Рис. 29. Этапы построения амплитудно-модулированного сигнала: блоки

После установки на схеме всех блоков из требуемых библиотек нужно выполнить соединение элементов схемы. Для соединения блоков необходимо указать курсором на «выход» блока, а затем нажать и, не отпуская левую клавишу мыши, провести линию к входу другого блока. После чего следует отпустить клавишу. Свидетельством того, что соединение создано, будет жирная стрелка у входного порта блока. Для создания точки разветвления в соединительной линии нужно подвести курсор к предполагаемому узлу и, нажав *правую* клавишу мыши, протянуть линию. Для удаления линии требуется ее выбрать, а затем нажать клавишу **Delete** на клавиатуре. Удерживая левую клавишу мыши, линию связи можно подвинуть, чтобы схема лучше читалась. Соединим все блоки и зададим их параметры, как описывалось выше.

Готовая структурная схема будет выглядеть, как показано на Рис. 30. Мы установили 3 входа у осциллографа, чтобы иметь возможность отдельно

наблюдать огибающую, модулированный сигнал и заполнение (соответственно сверху вниз).

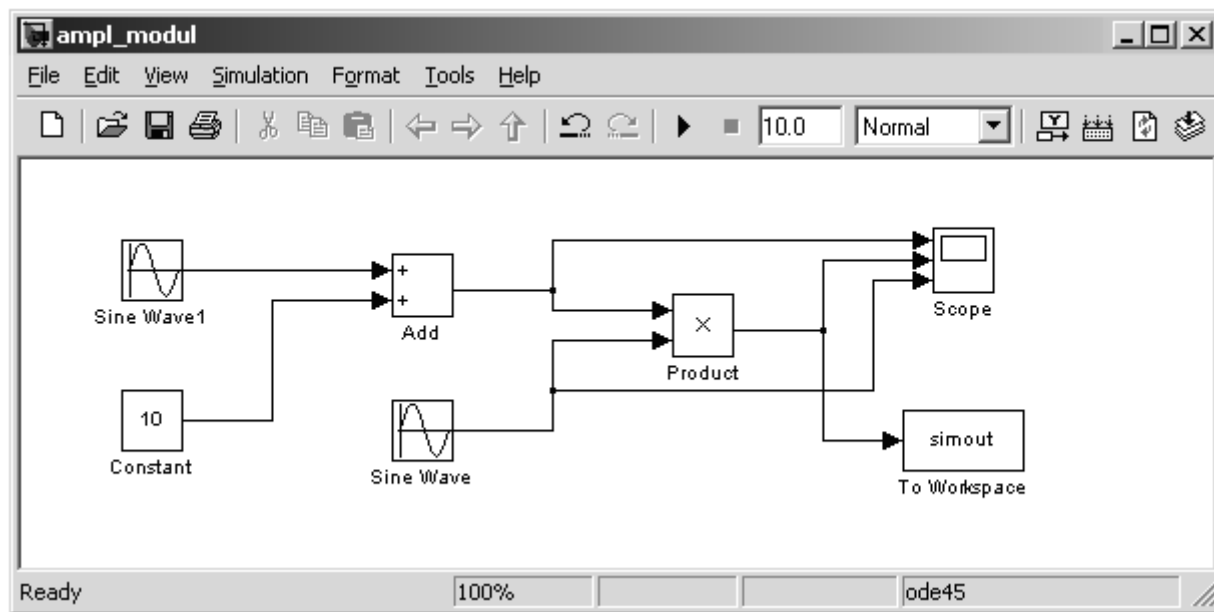


Рис. 30. Этапы построения амплитудно-модулированного сигнала: готовая Simulink-модель

После составления структурной схемы необходимо сохранить ее в виде файла на диске, выбрав пункт меню **File/Save As...** в окне схемы и указав папку и имя файла. Следует иметь в виду, что имя файла не должно превышать 32 символов, должно начинаться с буквы и не может содержать символы кириллицы и спецсимволы. Это же требование относится и к пути файла (к тем папкам, в которых сохраняется файл). При последующем редактировании схемы можно пользоваться пунктом меню **File/Save**. При повторных запусках программы **Simulink** загрузка схемы осуществляется с помощью меню **File/Open...** в окне обозревателя библиотеки или из основного окна **MATLAB**.

Для выполнения моделирования нужно нажать на кнопку команды **Start** панели инструментов рабочего окна модели или выбрать в меню **Simulation/ Start**. В этом случае программа выполнит моделирование с заданными по умолчанию параметрами. В окне осциллографа можно увидеть результат. В данном случае результат получится неудовлетворительным (Рис. 50), т. к. для этой задачи нельзя использовать параметры решения, заданные по умолчанию (более подробно параметры решения будут рассматриваться в следующем разделе).

Сейчас же, выполнив команду на панели управления меню **Simulation/Configuration Parameters ...**, в появившемся окне управления параметрами расчета изменим параметры **Max step size** и **Stop time**, как показано на Рис. 31.

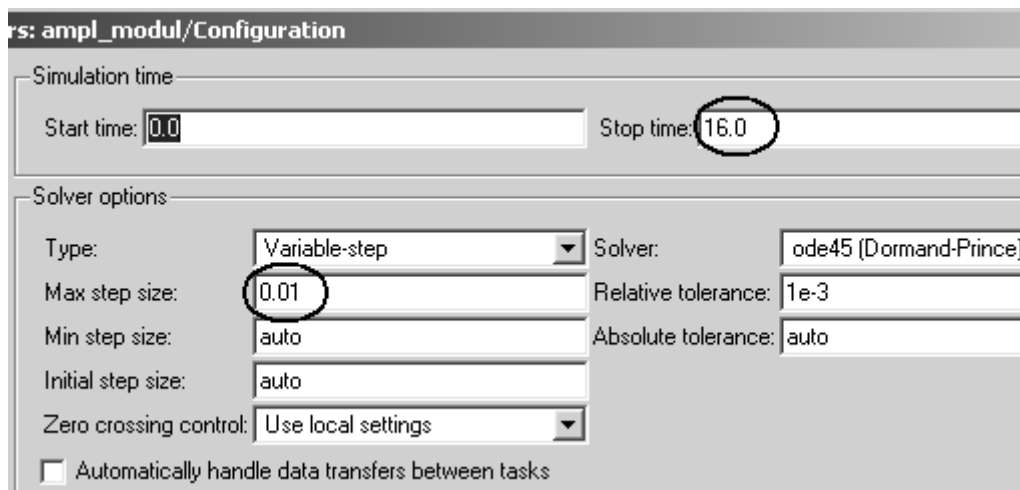


Рис. 31. Часть окна управления параметрами расчета

После выполнения расчета на экране осциллографа появятся результирующие сигналы (Рис. 32). На этом рисунке представлено три графика. Верхний – огибающая, нижний – заполнение и средний – модулированный сигнал.

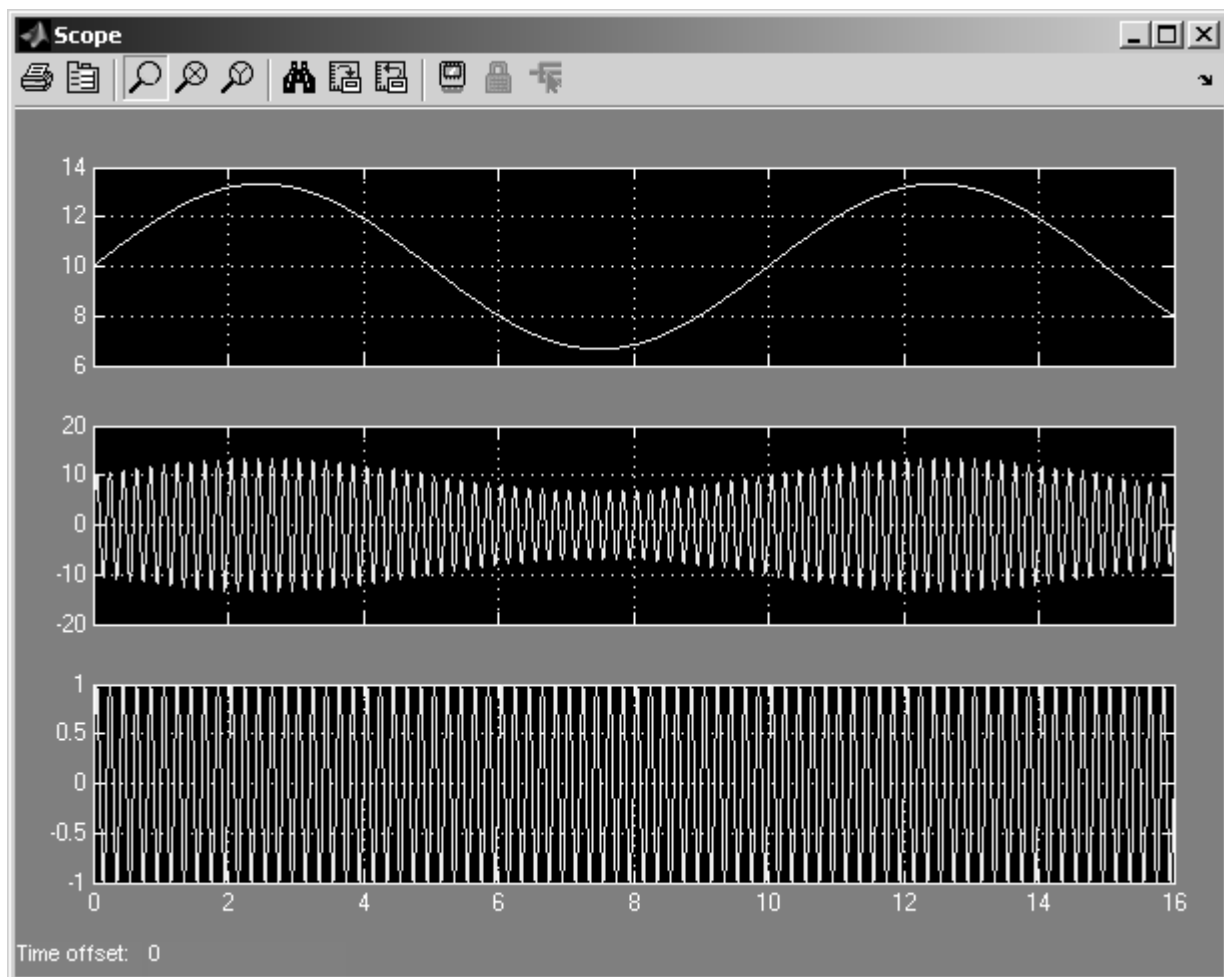


Рис. 32. Этапы построения амплитудно-модулированного сигнала: результаты моделирования

2.2.2. Генерирование импульсного сигнала

Для формирования сигналов сложной формы можно использовать комбинацию готовых блоков библиотеки **Sources** или сформировать сигнал с помощью команд **MATLAB** и воспользоваться блоками **From File** или **From Workspace**.

Приведем пример формирования сигнала единичной импульсной функции (δ -функции). Приложенная **Simulink**-модель для этого примера называется **pulse_signal.mdl**. Данный сигнал часто используется как тестовый сигнал для исследования поведения линейных систем. Единичный импульс $\delta(t-a)$ условно определяется как сигнал, имеющий нулевую (малую) длительность и обладающий свойствами:

$$\begin{aligned}\delta(t-a) &= 0, \quad t \neq a, \\ \int_{-\infty}^{\infty} \delta(t) dt &= 1.\end{aligned}\tag{3}$$

Можно приближенно реализовать δ -функцию двумя ступенчатыми сигналами. Рассмотрим схему на Рис. 33.

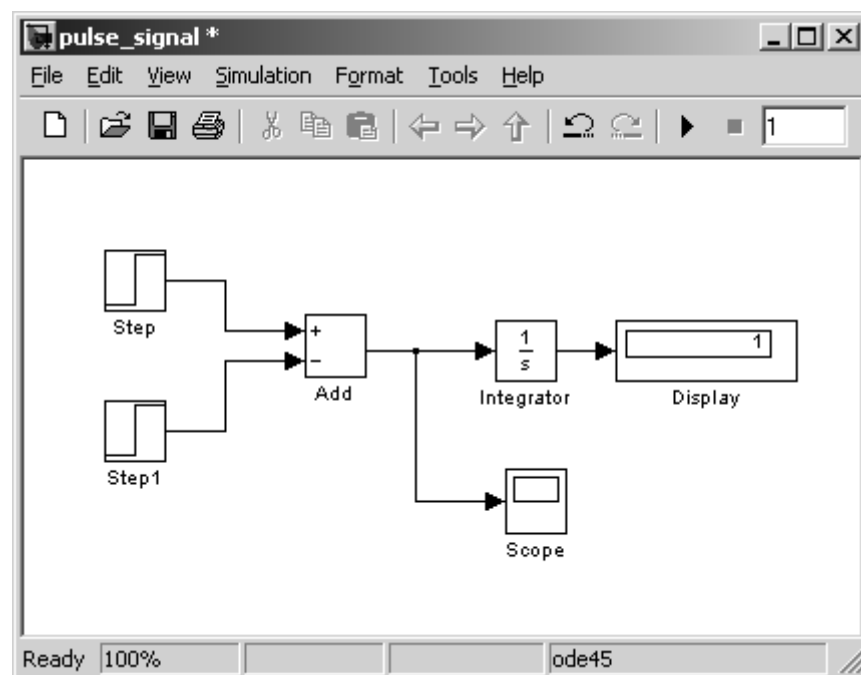


Рис. 33. Пример формирования сигнала единичной импульсной функции

В момент времени $t = 0,5$ с подается импульс амплитудой $A = 100$ и малой длительностью $\tau = 0,01$ с, так что величина $A \tau = 1$. Следует обратить внимание на задание величины τ , которая может определяться наибольшей собственной частотой в спектре модели. Импульсный сигнал формируется с помощью двух ступенек, одна из которых вычитается из другой. Параметры

блоков «Генератор ступенчатого сигнала» **Step** изображены на Рис. 34. На этой вкладке можно задать:

1. **Step time** – время наступления перепада сигнала (с),
2. **Initial value** – начальное значение сигнала,
3. **Final value** – конечное значение сигнала.

Величина ступеньки будет равна $|\text{Final value} - \text{Initial value}|$.

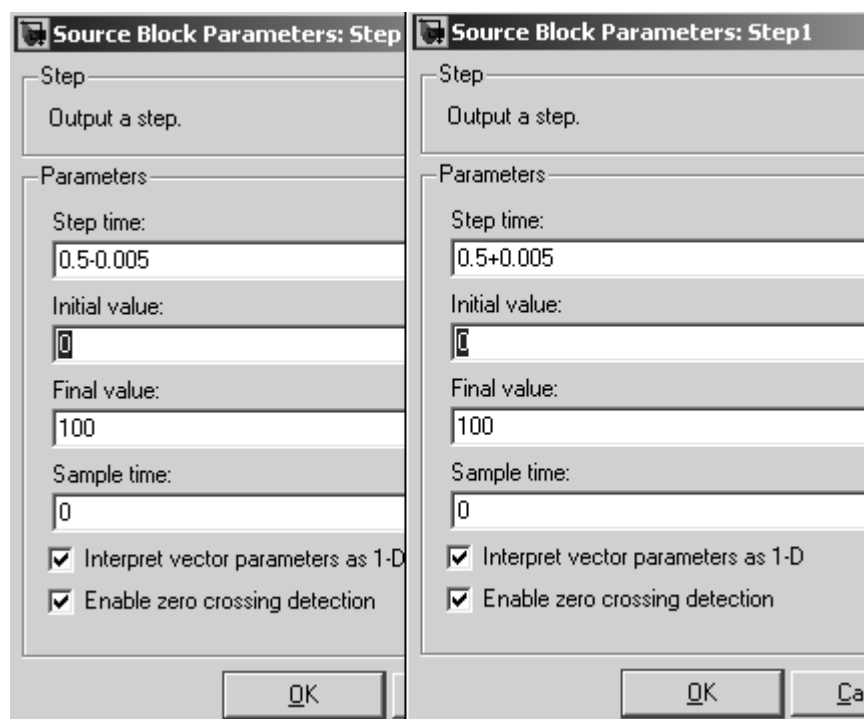


Рис. 34 Параметры блоков «Генератор ступенчатого сигнала» **Step**

Из библиотеки **Math Operations** – «Блоки математических операций» мы уже рассматривали блок вычисления суммы. Выставим последовательность знаков в этом блоке + -, что соответствует разности двух функций ступеньки.

После вычитания функций ступеньки, результирующий сигнал подается на блок «Осциллограф» и дальше передается этим блоком в рабочую область **MATLAB**. Результат моделирования представлен на Рис. 35.

Для проверки свойства нормировки δ -функции в схему включены еще два блока – «Интегратор» и «Цифровой дисплей». Блок интегрирования **Integrator** находится в библиотеке **Continuous** – «Аналоговые блоки» и выполняет интегрирование входного сигнала по формуле:

$$y(t) = y(t_0) + \int_{t_0}^t x(\tau) d\tau, \quad (4)$$

где $x(t)$ – входной сигнал, $y(t)$ – выходной сигнал, t_0 – начальное время, t – время моделирования.

Окно параметров этого блока изображено на Рис. 36.

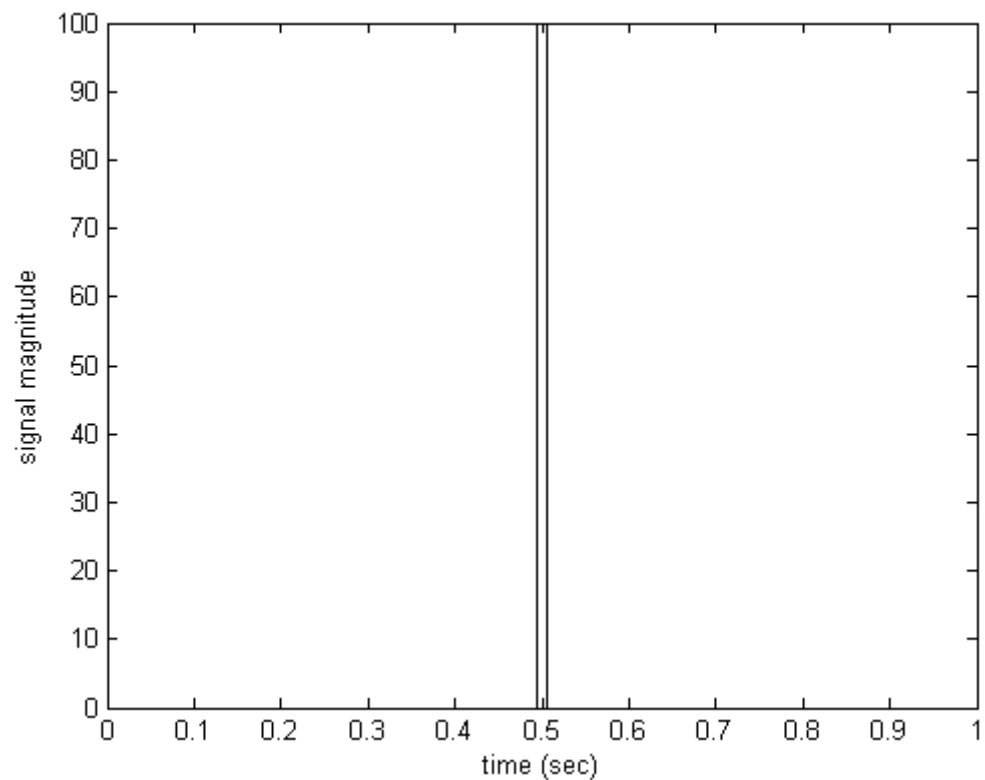


Рис. 35. Результат моделирования единичной импульсной функции

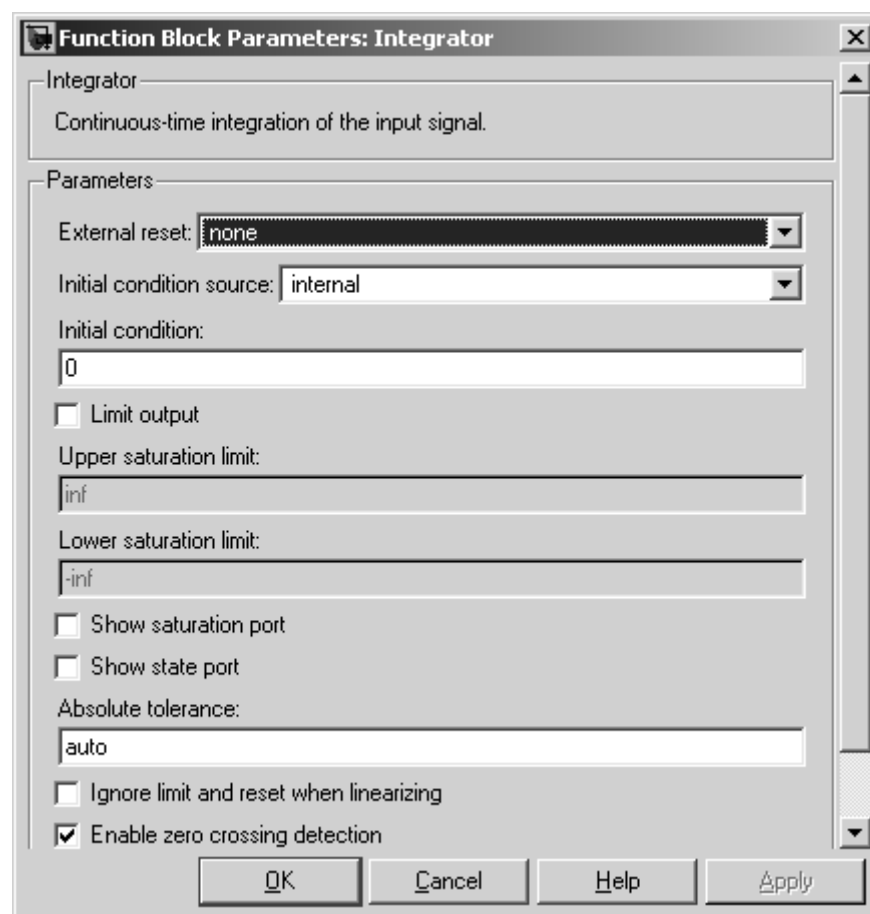


Рис. 36. Окно параметров блока «Интегратор»

В рассматриваемом примере в начальный момент времени t_0 выражение (4) равно нулю $y(t_0) = 0$, поэтому в окне параметра **Initial condition** (начальное условие) оставляем, заданный по умолчанию ноль.

Блок интегрирования, как и другие блоки, предусмотрен для использования и в других, более сложных ситуациях. Возможно использование внешнего сигнала для управления интегрированием, использование ограничения и др. Кратко перечислим параметры этого блока:

- **External reset** – внешний сброс. Тип внешнего управляющего сигнала, обеспечивающего сброс интегратора к начальному состоянию. Выбирается из списка:

1. **none** – нет (сброс не выполняется),
2. **rising** - нарастающий сигнал (передний фронт сигнала),
3. **falling** - спадающий сигнал (задний фронт сигнала),
4. **either** – нарастающий, либо спадающий сигнал,
5. **level** – не нулевой сигнал (сброс выполняется, если сигнал на управляющем входе становится не равным нулю).

В том случае, если выбран какой-либо (но не **none**), тип управляющего сигнала, то на изображении блока появляется дополнительный управляющий вход. Рядом с дополнительным входом будет показано условное обозначение управляющего сигнала.

- **Initial condition source** – источник начального значения выходного сигнала. Выбирается из списка:

1. **internal** – внутренний,
2. **external** – внешний. В этом случае на изображении блока появляется дополнительный вход, обозначенный x_0 , на который необходимо подать сигнал, задающий начальное значение выходного сигнала интегратора.

- **Initial condition** – начальное условие. Установка начального значения выходного сигнала интегратора. Параметр доступен, если выбран внутренний источник начального значения выходного сигнала.

- **Limit output** (флажок) – использование ограничения выходного сигнала.

- **Upper saturation limit** – верхний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf**, то есть $+\infty$.

- **Lower saturation limit** – нижний уровень ограничения выходного сигнала. Может быть задан как числом, так и символьной последовательностью **inf**, то есть $-\infty$.

- **Show saturation port** – управляет отображением порта, выводящего сигнал, свидетельствующий о выходе интегратора на ограничение. Выходной сигнал данного порта может принимать следующие значения:

1. **0**, если интегратор не находится на ограничении.
2. **+1**, если выходной сигнал интегратора достиг верхнего ограничивающего предела.
3. **-1**, если выходной сигнал интегратора достиг нижнего ограничивающего предела.

- **Show state port** (флажок) – отобразить/скрыть порт состояния блока. Данный порт используется в том случае, если выходной сигнал интегратора требуется подать в качестве сигнала обратной связи этого же интегратора. Например, при установке начальных условий через внешний порт или при сбросе интегратора через порт сброса. Выходной сигнал с этого порта может использоваться также для организации взаимодействия с управляемой подсистемой.

- **Absolute tolerance** – абсолютная погрешность.

Для визуализации результата используется блок «Цифровой дисплей» **Display**. Он отображает текущее значение сигнала в виде числа. Окно параметров этого блока изображено на Рис. 37.

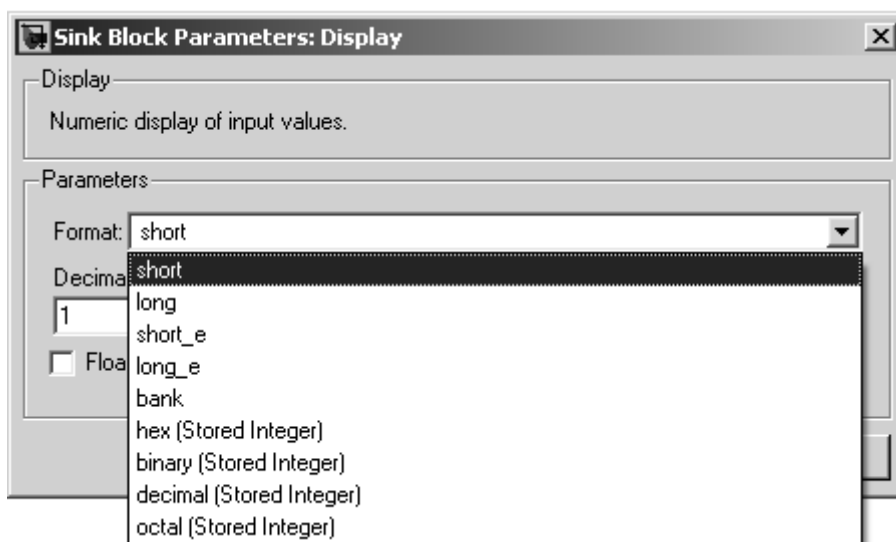


Рис. 37. Окно параметров блока «Цифровой дисплей»

В выпадающем списке окна **Format** можно выбрать формат отображения данных:

1. **short** – 5 значащих десятичных цифр.
2. **long** – 15 значащих десятичных цифр.

3. **short_e** – 5 значащих десятичных цифр и 3 символа степени десяти.
4. **long_e** – 15 значащих десятичных цифр и 3 символа степени десяти.
5. **bank** – «денежный» формат. Формат с фиксированной точкой и двумя десятичными цифрами в дробной части числа.
6. **hex** – шестнадцатеричная система счисления.
7. **binary** – двоичная система счисления.
8. **decimal** – десятичная система счисления.
9. **octal** – восьмеричная система счисления.

Перечислим другие параметры этого блока:

- **Decimation** – кратность отображения входного сигнала.

При **Decimation = 1** отображается каждое значение входного сигнала, при **Decimation = 2** отображается каждое второе значение, при **Decimation = 3** – каждое третье значение и т.д.

- **Sample time** – шаг модельного времени. Определяет дискретность отображения данных.

- **Floating display** (флажок) – перевод блока в «свободный» режим. В данном режиме входной порт блока отсутствует, а выбор сигнала для отображения выполняется щелчком левой клавиши мыши на соответствующей линии связи. В этом режиме для параметра расчета **Signal storage reuse** должно быть установлено значение **off** (вкладка **Advanced** в окне диалога **Simulation parameters...**).

2.2.3. Использование блока From Workspace

Рассмотрим пример использования блока «Считывания данных из рабочего пространства» **From Workspace**. Создадим выполняемый файл **MATLAB** (в приложении файл называется **from_workspace_mat.m**):

```
t=0:0.01:10;      %формируем вектор отсчетов по времени
tau=5;           %длительность прямоугольного импульса
y1=t<tau;        %формируем прямоугольный импульс
y2=sin(2*pi*10/5*t); %формируем синусоидальный сигнал
y=y1.*y2;        %перемножение
plot(t,y)        %построение графика
simin=[t' y'];    %создание массива simin
```

Этот файл формирует сигнал радиоимпульса: огибающая – прямоугольный импульс длительностью τ с синусоидальным заполнением с частотой $10/\tau$ (Рис. 38).

Сохраним этот m-файл в рабочей папке **MATLAB** и выполним его. Результатом выполнения будет построение графика (Рис. 38) и создание в рабочей области **MATLAB** массива **simin**.

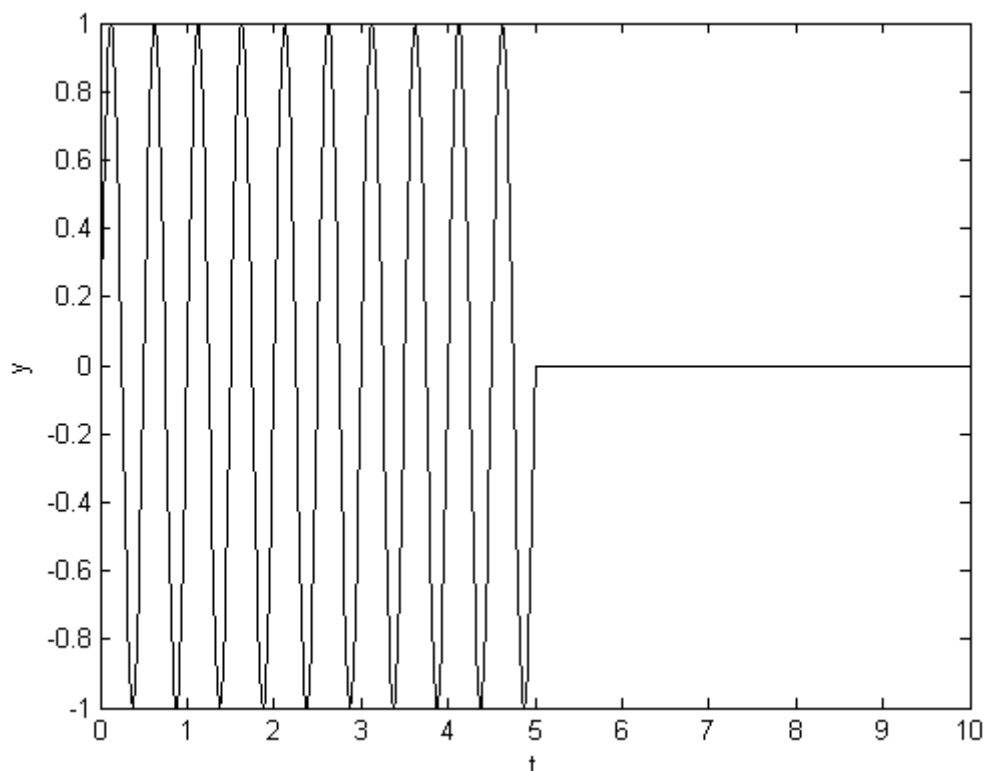


Рис. 38. Радиоимпульс

Создадим рабочую схему в **Simulink**, как показано на Рис. 39. Название модели в приложении **from_workspace.mdl**. Заметим, что имена m-файла и **Simulink**-модели должны быть различны.

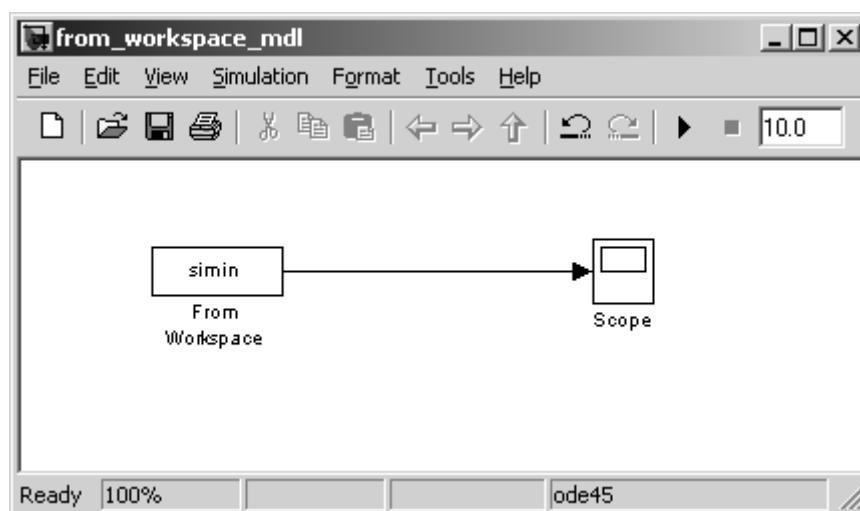


Рис. 39. Использование «Блока считывания данных из рабочего пространства» From Workspace

Блок **From Workspace** может считывать данные только в виде массива или структуры со временем, поэтому в рабочей области **MATLAB** мы сформировали массив **simin**. В окне параметров блока **From Workspace** имя переменной оставим **simin** и укажем шаг изменения входного сигнала (Рис. 40)

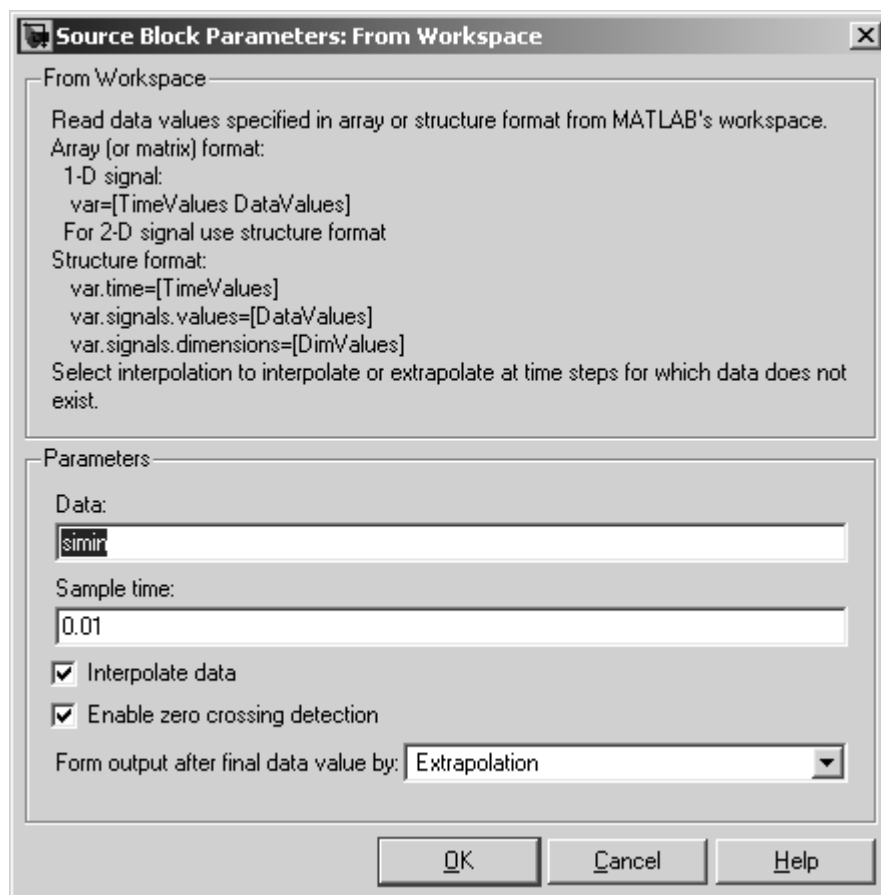


Рис. 40. Окно параметров блока From Workspace

Параметры блока **From Workspace**:

1. **Data** – имя переменной (матрицы или структуры) содержащей данные.
2. **Sample time** - шаг изменения выходного сигнала блока.
3. **Interpolate data** — интерполяция данных для значений модельного времени не совпадающих со значениями в переменной **Data**.
4. **Form output after final data value by** – вид выходного сигнала по окончании значений времени в переменной **Data**:
 - **Extrapolate** – линейная экстраполяция сигналов,
 - **SettingToZero** – нулевые значения сигналов,
 - **HoldingFinalValue** – выходные значения сигналов равны последним значениям,
 - **CyclicRepetition** – циклическое повторение значений сигналов.

Данный вариант может использоваться, только если переменная **Data** имеет формат **Structure without time**.

Вместо строчки создания массива в m-файле можно написать несколько строчек создания структуры со временем:

```
%создание структуры simin:
simin.time=[t']; %время
simin.signals.values=[y']; %значения
simin.signals.dimensions=[1]; %размерность
hold on
plot(simin.time, simin.signals.values, 'r')
```

В обоих случаях зарегистрированный осциллографом сигнал должен совпадать с сигналом, изображенным на Рис. 38.

2.2.4. Использование передаточной функции Transfer Fcn

Рассмотрим интегрирующую цепочку (или фильтр низких частот) (Рис. 41), состоящую из конденсатора с емкостью C и резистора R .

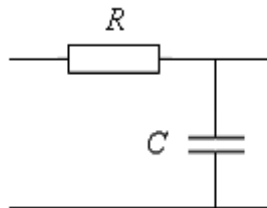


Рис. 41. Фильтр низких частот

Отношение спектральной функции выходного сигнала к спектральной функции входного сигнала называется **передаточной функцией** (или **коэффициентом передачи**). Значение коэффициента передачи линейной системы для заданного значения частоты ω_0 может быть получено, если на вход такой системы подавать гармонический сигнал $\dot{x}(t) = A_1 e^{i\omega_0 t}$ (в последующем точками над переменными мы будем обозначать комплексные величины, чтобы отличать их от действительных величин, измеряемых на практике). Тогда выходной сигнал является также гармоническим $\dot{y}(t) = A_2 e^{i(\omega_0 t + \varphi)}$. В этом случае коэффициент передачи будет:

$$\dot{K} = \frac{A_2}{A_1} e^{i\varphi} \quad (5)$$

Модуль передаточной функции выражает отношение амплитуд сигналов, и его зависимость от частоты называется амплитудно-частотной характеристикой (АЧХ), а аргумент – фазовый сдвиг между сигналами, и его зависимость от частоты называется фазо-частотной характеристикой (ФЧХ).

Линейная система (с нулевыми начальными значениями) полностью описывается таким образом определенной передаточной функцией. Блок «Передаточной функции» в **Simulink** называется **Transfer Fcn** (пиктограмма блока изображена на Рис. 42) и находится в библиотеке **Continuous**.

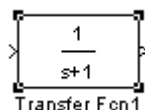


Рис. 42. Пиктограмма блока «Передаточной функции»

Рассмотрим электрическую схему, состоящую из интегрирующей цепочки, на вход которой подается последовательность прямоугольных импульсов. Имя этой модели в приложении **transfer_fcn.mdl**. Выходной сигнал регистрируется осциллографом. Модель этой схемы в **Simulink** будет выглядеть, как показано на Рис. 43.

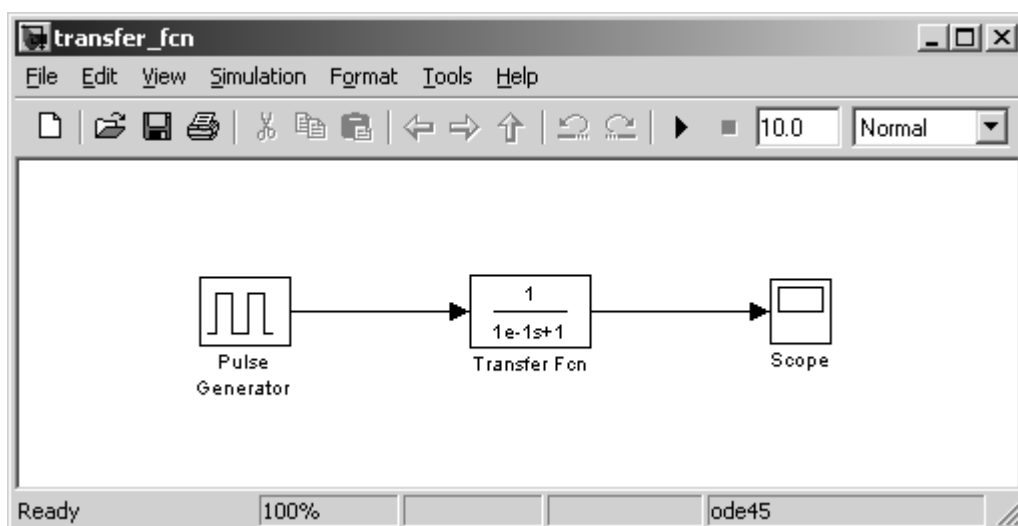


Рис. 43. Модель интегрирующей цепочки

Выберем следующие параметры прямоугольных импульсов для блока **Pulse Generator**:

Амплитуда = 1

Период = 3 с

Скважность = 50%

Фазовая задержка = 0 с.

Передаточная функция интегрирующей цепочки выражается как:

$$\dot{K} = \frac{1}{i\omega\tau + 1} = \frac{1}{\tau s + 1}, \quad (6)$$

где $\tau = RC$ – постоянная времени цепочки,
 ω – циклическая частота,
 $s = i\omega$ – переменная Лапласа.

В блоке **Simulink** передаточная функция задается векторами (*num* и *den*), состоящими из коэффициентов числителя и знаменателя в порядке уменьшения степени s :

$$\dot{K}(s) = \frac{Y(s)}{X(s)} = \frac{num(s)}{den(s)} = \frac{num(1)s^{n-1} + num(2)s^{n-2} + \dots + num(n)}{den(1)s^{d-1} + den(2)s^{d-2} + \dots + den(d)} \quad (7)$$

Например, для интегрирующей цепочки вектор коэффициентов числителя состоит из одного элемента $num(s) = [1]$, а знаменателя – из двух элементов $den(s) = [\tau \ 1]$. Выберем $\tau = 0,1$ с, тогда окно параметров для передаточной функции будет выглядеть, как показано на Рис. 44.

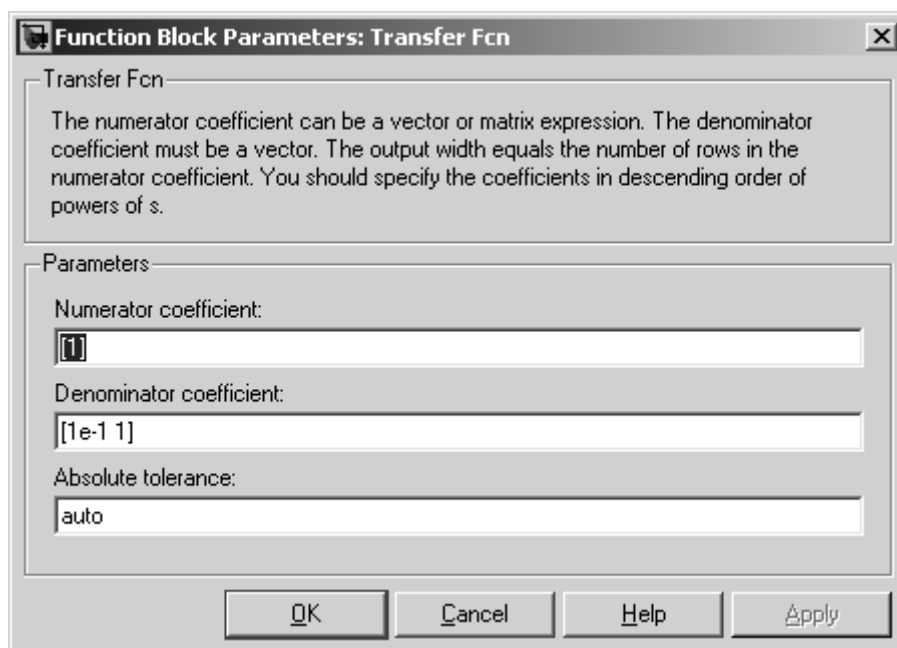


Рис. 44. Окно параметров блока передаточной функции

Искажение прямоугольных импульсов интегрирующей цепочкой показано на Рис. 45.

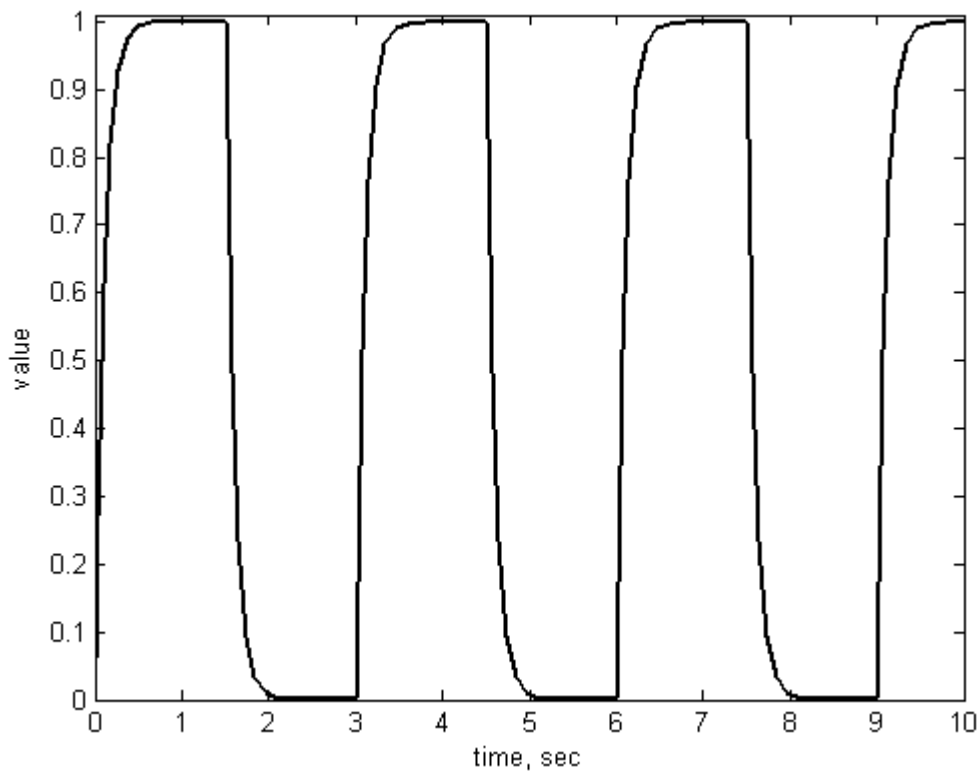


Рис. 45. Прямоугольный импульс после прохождения через интегрирующую цепочку

2.2.5. Модель системы, заданная дифференциальным уравнением

В предыдущем примере была описана модель электрической системы (интегрирующей цепочки), заданная с помощью передаточной функции. Другой способ описания физических систем – дифференциальное уравнение. Рассмотрим колебательный контур, последовательно с которым включен источник синусоидального переменного напряжения (Рис. 46).

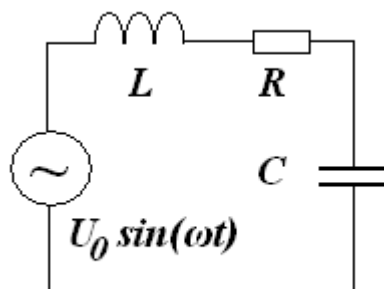


Рис. 46. Последовательный колебательный контур

По второму закону Кирхгофа имеем:

$$U_0 \sin(\omega t) = U_R + U_L + U_C = RI + L \frac{dI}{dt} + \frac{1}{C} \int_0^t I dt. \quad (8)$$

Продифференцировав это выражение по времени, получим неоднородное линейное дифференциальное уравнение с постоянными коэффициентами второго порядка, описывающее вынужденные изменение тока в колебательном контуре:

$$\frac{d^2 I}{dt^2} + 2\beta \frac{dI}{dt} + \omega_0^2 I = A \cos \omega t, \quad (9)$$

где

$\omega_0 = \sqrt{\frac{1}{LC}}$ – собственная, частота колебательного контура,

$\beta = \frac{R}{2L}$ – коэффициент затухания,

$A = \frac{\omega U_0}{L}$.

В качестве начальных условий при $t_0 = 0$ выберем:

$$I(t_0) = 0A$$

$$I'(t_0) = 1A/c$$

Модель этого дифференциального уравнения может быть представлена в виде структурной схемы в **Simulink** с использованием блока «Интегратор». Принцип использования блока «Интегратор» проиллюстрирован на Рис. 47. Если входная величина блока будет второй производной от некоторой величины по времени, то на выходе получается первая производная (с учетом начальных условий).

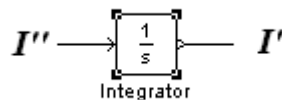


Рис. 47. Использование блока «Интегратор» для моделирования дифференциальных уравнений

Математически эту операцию можно записать как:

$$I'(t) = I'(t_0) + \int_{t_0}^t I''(\tau) d\tau,$$

где $I''(t)$ – входной для блока «Интегратор» сигнал, $I'(t)$ – выходной сигнал, t_0 – начальное время, t – время окончания моделирования. Из этого уравнения видно, что для блока «Интегратор» в параметрах (**Initial condition**) необходимо задать начальные условия интегрирования $I'(t_0) = 1A/c$.

Последующее использование блока интегрирования для первой производной позволяет получить саму величину I :

$$I(t) = I(t_0) + \int_{t_0}^t I'(\tau) d\tau$$

В этом случае нулевые начальные условия интегрирования $I(t_0) = 0A$.

Перепишем уравнение колебательного контура в виде:

$$I'' = -2\beta I' - \omega_0^2 I + A \cos \omega t$$

На Рис. 48 изображена схема **Simulink**, соответствующая этому уравнению. В приложении эта модель называется **diff_equ.mdl**.

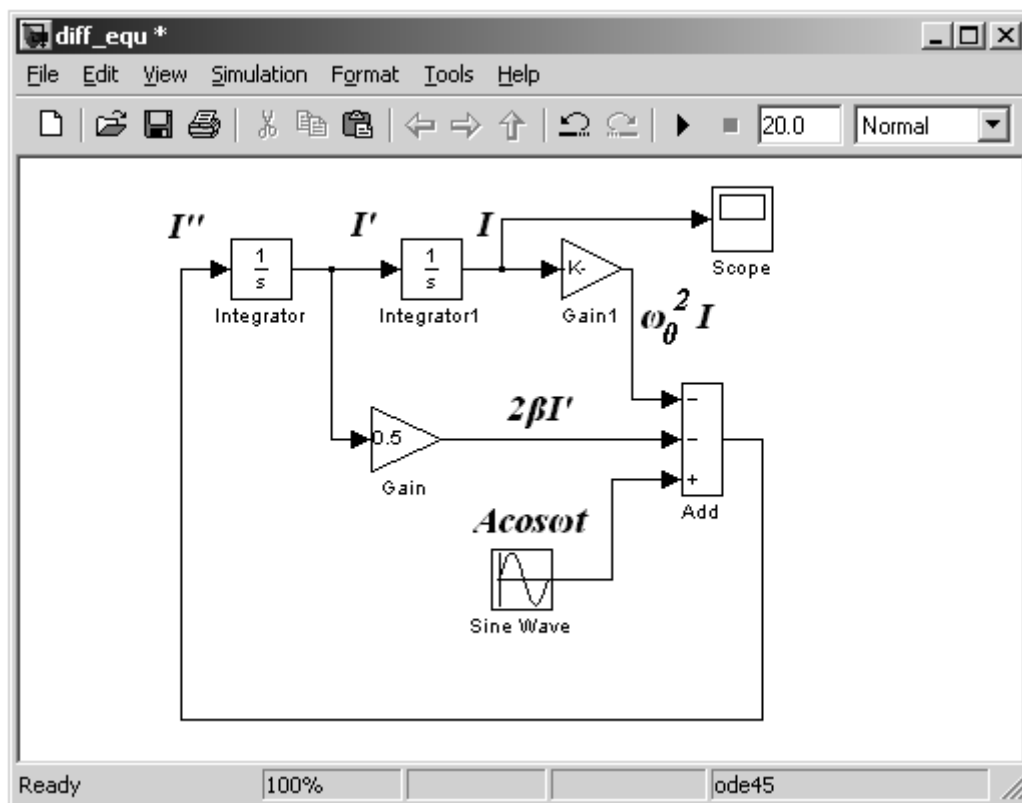


Рис. 48. Модель колебательного контура с синусоидальным источником напряжения

В зависимости от параметров контура и частоты источника возможно различное поведение такой системы. Для частного случая $\omega_0 = 2\pi \cdot 10^5$ рад/с, $2\beta = 10^5$ Ом/Гн, $A = (2\pi)^2 \cdot 10^{11}$ А/с² и $\omega = 2\pi \cdot 10^6$ рад/с зависимость тока от

времени изображена на Рис. 49 (в верхнем правом углу показан увеличенный участок этого графика). Такие параметры уравнения возможны, например, для следующих значений параметров электрической цепи: $C = 10^{-7}/(2\pi)$ Ф, $L = 10^{-3}/(2\pi)$ Гн, $R = 100/(2\pi)$ Ом и $U_0 = 100$ В.

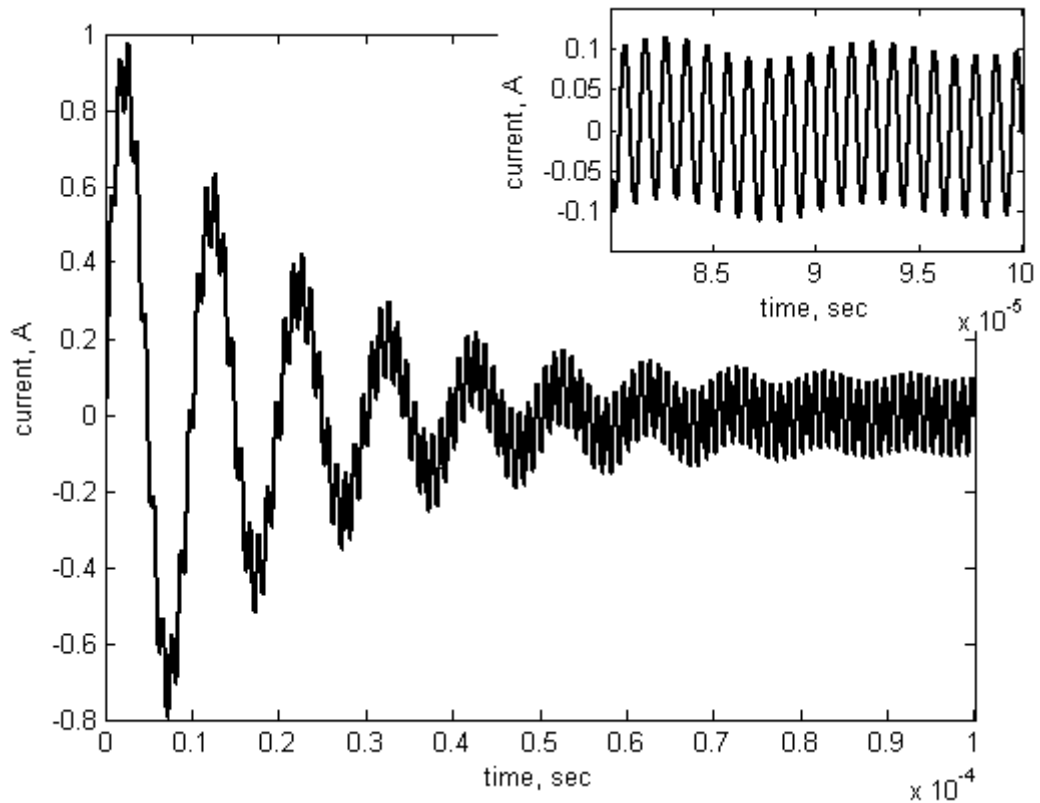


Рис. 49. Установление вынужденных колебаний тока в колебательном контуре

2.3. Установка параметров расчета и его выполнение

В примере пункта 2.2.1, используя значения параметров моделирования принятые по умолчанию, мы получили искаженный сигнал (Рис. 50). Это объясняется тем, что решатель не всегда правильно определяет требуемый шаг по времени, поэтому его желательно ограничивать с помощью параметра **Max Step**. В нашем случае частоты сигналов заполнения и огибающей сильно отличаются друг от друга $\frac{\omega_0}{\Omega} = 50$, а настройки, установленные по умолчанию, для данной частоты заполнения позволяют выводить меньше одной точки на период, что, естественно, недостаточно для корректного изображения сигнала.

Поэтому, перед выполнением расчетов необходимо задать параметры расчета. Задание параметров выполняется в панели управления меню **Simulation/Configuration Parameters**

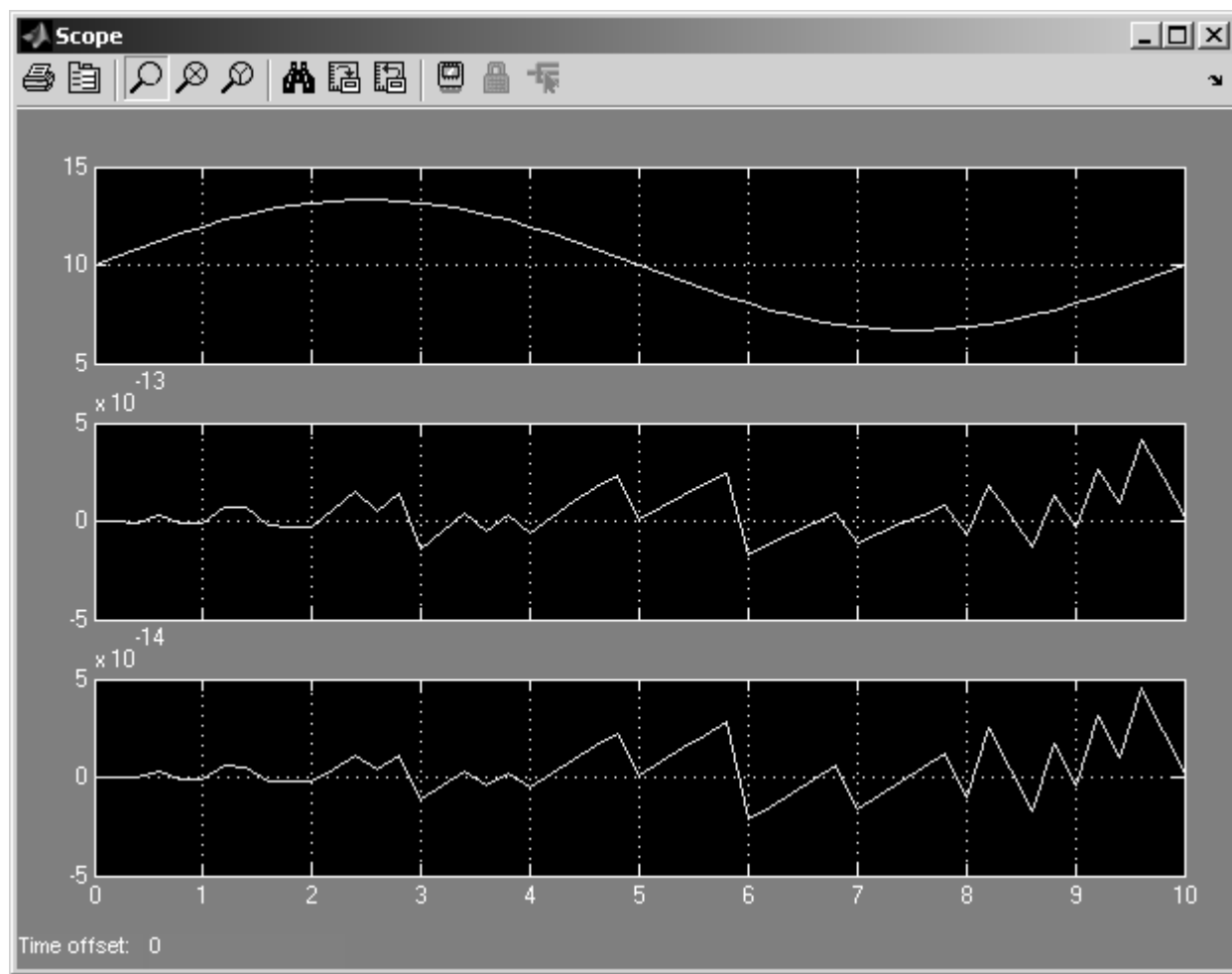


Рис. 50. Пример искаженного за счет большого шага по времени сигнала

Вид панели управления приведен на Рис. 51. Слева на панели управления параметрами расчета находится список вкладок с элементами управления в виде дерева. Перечислим некоторые из них:

- **Solver** (Расчет) – установка параметров расчета модели.
- **Data Import/Export** (Ввод/вывод данных в рабочую область) – установка параметров обмена данными с рабочей областью **MATLAB**.
- **Diagnostics** (Диагностика) – выбор параметров диагностического режима.

Мы рассмотрим только параметры, регулируемые на вкладке **Solver**. Установка параметров расчета модели выполняется с помощью элементов управления, размещенных на вкладке **Solver**. Эти элементы разделены на группы (Рис. 51): **Simulation time** (Интервал моделирования или, иными словами, время расчета) и **Solver options** (Параметры расчета).

Рассмотрим подробнее эти параметры:

- **Simulation time** (Интервал моделирования)

Время расчета задается указанием начального (**Start time**) и конечного (**Stop time**) значений времени расчета. По умолчанию эти величины задаются

равными 0 и 10 с, соответственно. Величина конечного времени должна выбираться пользователем, исходя из условий решаемой задачи.

- **Solver options** (Параметры расчета)

При выборе параметров расчета необходимо указать способ моделирования (**Type**) и метод расчета нового состояния системы. Раскрывающийся список окна параметра **Type** показан на Рис. 52. Для этого параметра доступны два варианта – с фиксированным (**Fixed-step**) или с переменным (**Variable-step**) шагом. Как правило, **Variable-step** используется для моделирования непрерывных систем, а **Fixed-step** – для дискретных.

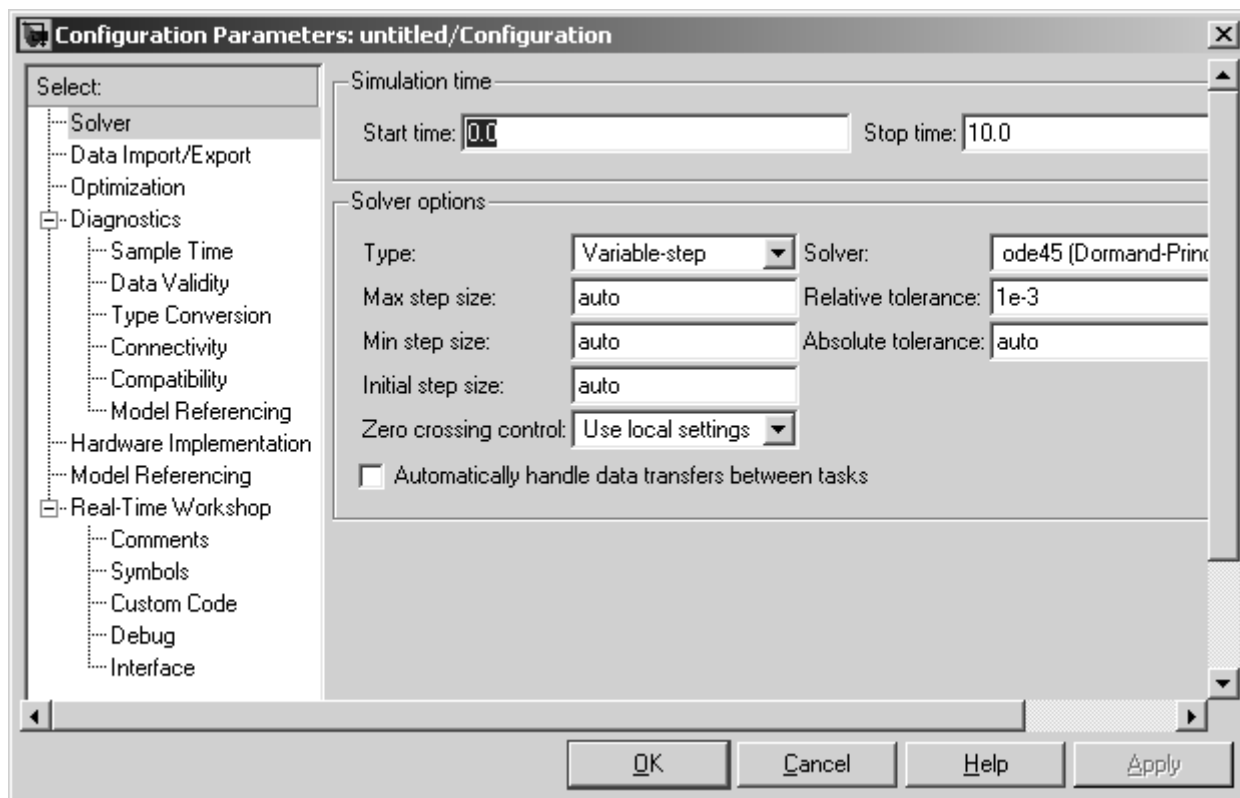


Рис. 51. Панель управления параметрами расчета

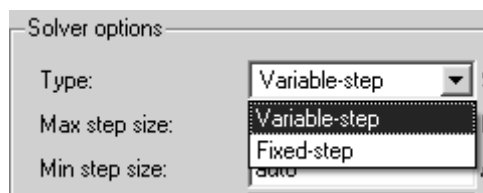


Рис. 52. Раскрывающийся список окна параметра **Type**

Список методов расчета нового состояния системы (окна **Solver**) содержит несколько вариантов (Рис. 53). Первый вариант (**discrete**) используется для расчета дискретных систем. Остальные методы используются для расчета непрерывных систем. Эти методы различны для переменного (**Variable-step**) и для фиксированного (**Fixed-step**) шага времени, но, по сути, представляют собой различные алгоритмы решения систем дифференциальных урав-

нений. Подробное описание каждого из методов расчета состояний системы приведено во встроенной справочной системе **MATLAB**. По умолчанию для переменного шага системой выбирается функция **ode45**. Это эффективный одношаговый решатель общего назначения, в котором реализуются явные методы Рунге-Кутты.

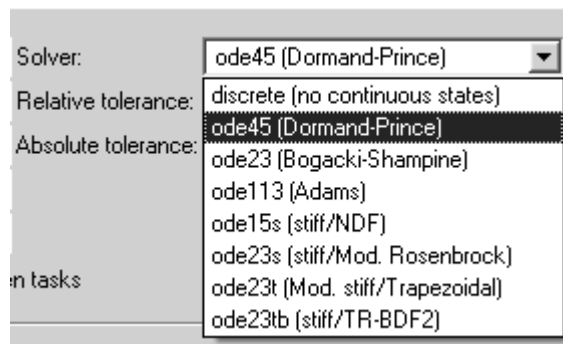


Рис. 53. Раскрывающийся список окна Solver

Ниже двух раскрывающихся списков **Type** находится область, содержание которой меняется в зависимости от выбранного способа изменения модельного времени. При выборе **Fixed-step** (Рис. 54) в данной области появляется текстовое поле **Fixed-step size** (величина фиксированного шага), позволяющее указывать величину шага моделирования. Величина шага моделирования по умолчанию устанавливается системой автоматически (**auto**). Требуемая величина шага может быть введена вместо значения **auto** либо в форме числа, либо в виде вычисляемого выражения (то же самое относится и ко всем параметрам, устанавливаемым системой автоматически).

При выборе **Fixed-step** необходимо также задать режим расчета (**Tasking mode for periodic sample times**). Для этого параметра доступны три варианта:

- **MultiTasking** (Многозадачный) – необходимо использовать, если в модели присутствуют параллельно работающие подсистемы, и результат работы модели зависит от временных параметров этих подсистем. Режим позволяет выявить несоответствие скорости и дискретности сигналов, пересылаемых блоками друг другу.
- **SingleTasking** (Однозадачный) - используется для тех моделей, в которых недостаточно строгая синхронизация работы отдельных составляющих не влияет на конечный результат моделирования.
- **Auto** (Автоматический выбор режима) - позволяет **Simulink** автоматически устанавливать режим **MultiTasking** для тех моделей, в которых используются блоки с различными скоростями передачи сигналов и режим **SingleTasking** для моделей, в которых содержатся блоки, оперирующие одинаковыми скоростями.

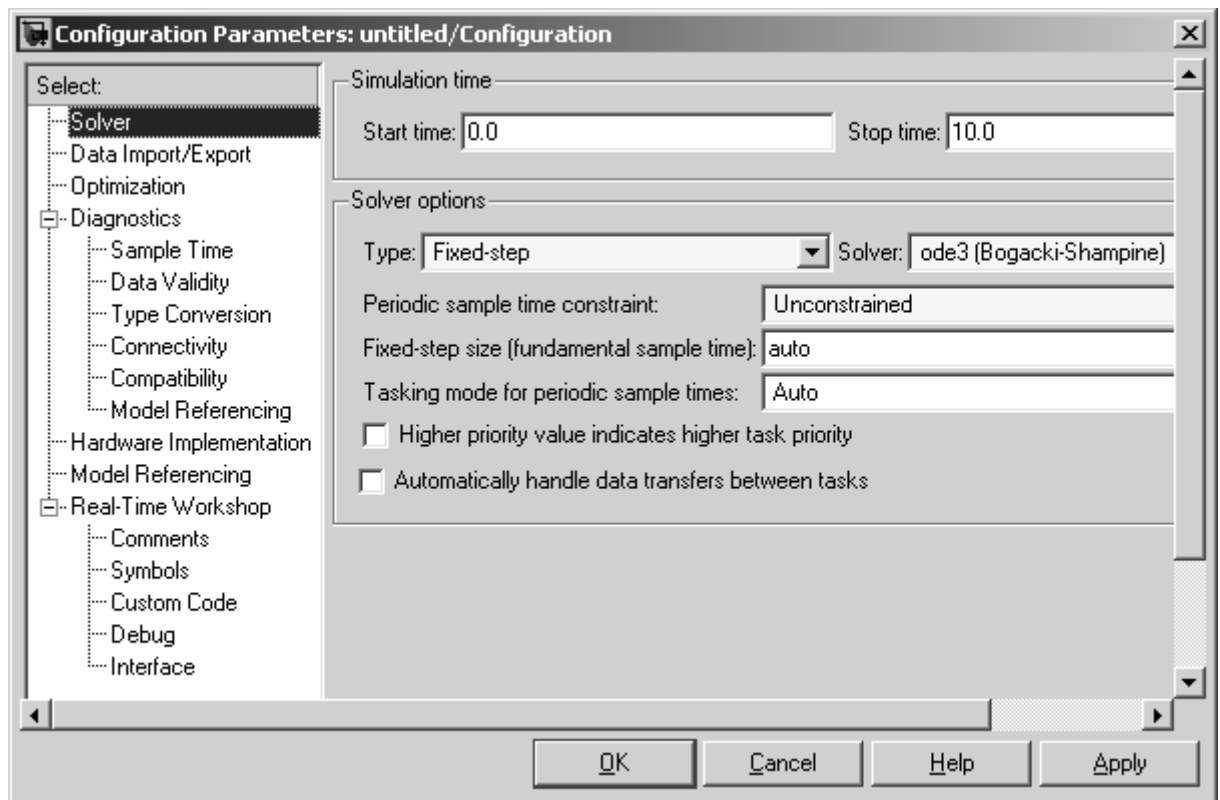


Рис. 54. Вкладка Solver при выборе фиксированного шага расчета

При выборе **Variable-step** в области появляются поля для установки трех параметров (Рис. 51):

- **Max step size** - максимальный шаг расчета. По умолчанию он устанавливается автоматически (**auto**) и его значение в этом случае равно $(\text{StopTime} - \text{StartTime})/50$. Довольно часто это значение оказывается слишком большим (как в рассмотренном выше примере), и наблюдаемые графики представляют собой ломаные (а не плавные) линии. В этом случае величину максимального шага расчета необходимо задавать явным образом.
- **Min step size** - минимальный шаг расчета.
- **Initial step size** - начальное значение шага моделирования.

При моделировании непрерывных систем с использованием переменного шага необходимо указать точность вычислений: относительную (**Relative tolerance**) и абсолютную (**Absolute tolerance**). По умолчанию они равны соответственно 10^{-3} и **auto**.





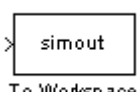
2.4. Основные блоки Simulink

Мы уже встречались с некоторыми блоками **Simulink**, сейчас в таблице перечислим назначение некоторых наиболее часто используемых блоков.




Блоки из библиотеки **Sources** (источники сигналов) имеют один выходной порт.


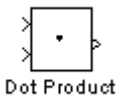
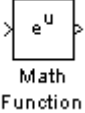
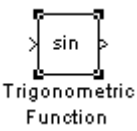
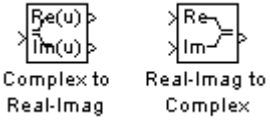
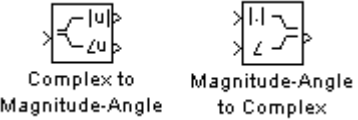

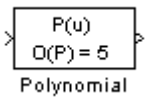
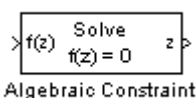

Sources – источники сигналов и воздействий	
 Constant	Источник постоянного сигнала Constant – задает постоянный по уровню сигнал. Значение константы может быть действительным или комплексным числом, вычисляемым выражением, вектором или матрицей.
 Sine Wave	Источник синусоидального сигнала Sine Wave – формирует синусоидальный сигнал с заданной частотой, амплитудой, фазой и смещением.
 Ramp	Источник линейно изменяющегося воздействия Ramp – формирует линейный сигнал вида $y = \text{наклон} * \text{время}$. Можно задавать время начала сигнала.
 Step	Генератор ступенчатого сигнала Step – формирует ступенчатый сигнал. Значения начального и конечного уровней могут быть не только положительными, но и отрицательными.
 Repeating Sequence	Источник пилообразного сигнала Repeating Sequence – формирует периодический пилообразный сигнал, заданной амплитуды и периода.
 Pulse Generator	Источник прямоугольного импульсного сигнала Pulse Generator – формирует прямоугольные импульсы, заданной амплитуды, периода и скважности.
 Chirp Signal	Генератор сигнала с линейно изменяющейся частотой Chirp Generator – формирует синусоидальные колебания, частота которых линейно изменяется.
 Uniform Random Number	Источник случайного сигнала с равномерным распределением Uniform Random Number – формирует случайный сигнал (в заданном диапазоне) с равномерным распределением.
 Random Number	Источник случайного сигнала с нормальным распределением Random Number – формирует случайный сигнал с нормальным (Гауссовым) распределением уровня сигнала.
 Band-Limited White Noise	Генератор белого шума Band-Limited White Noise – создает сигнал заданной мощности, равномерно распределенной по частоте.
 Clock	Источник временного сигнала Clock – формирует сигнал, величина которого на каждом шаге расчета равна текущему времени моделирования.
 From File	Блок считывания данных из файла From File – получение данных из внешнего файла.
 From Workspace	Блок считывания данных из рабочего пространства From Workspace – получение данных из рабочего пространства MATLAB.


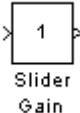

Блоки библиотеки **Sinks** (приемники сигналов) используются для визуализации и записи полученных результатов. Эти блоки могут использоваться для отображения не только скалярных сигналов, но также векторных, матричных и комплексных.

Sinks - приемники сигналов	
 Scope	Осциллограф Scope – строит графики исследуемых сигналов как функцию времени.
 XY Graph	Графопостроитель XY Graph – строит график одного сигнала как функцию другого (график вида $Y(X)$).
 Display	Цифровой дисплей Display – отображает значение сигнала в виде числа.
 To File	Блок сохранения данных в файле To File – блок записывает данные, поступающие на его вход, в файл. Данные в файле сохраняются в виде матрицы. Значения времени записываются в первой строке матрицы, а в остальных строках будут находиться значения сигналов, соответствующих данным моментам времени.
 To Workspace	Блок сохранения данных в рабочей области To Workspace – блок записывает данные, поступающие на его вход, в рабочую область MATLAB .



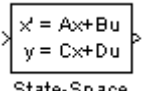
Блоки библиотеки математических операций **Math**. Для всех блоков поддерживаются операции с матричными величинами и комплексными числами.

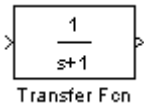
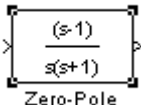

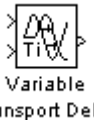
Math – блоки математических операций	
 Abs	Блок вычисления модуля Abs – вычисляет абсолютное значение величины сигнала. Этот блок может использоваться также для вычисления модуля сигнала комплексного типа.
 Add	Блок вычисления суммы Add – выполняет вычисление суммы (или разности) текущих значений сигналов. Форма блока может меняться с прямоугольной на круглую. Число входов и математические операции также могут меняться.
 Divide	Блок деления Divide – выполняет вычисление отношения (или произведения) текущих значений сигналов. Число входов и математические операции могут меняться.

	<p>Блок усилителя Gain – выполняет умножение входного сигнала на константу.</p>
	<p>Блок скалярного умножения Dot Product – выполняет вычисление скалярного произведения двух векторов.</p>
	<p>Блок вычисления математических функций Math Function – выполняет вычисление некоторых математических функций, которые можно выбрать из списка.</p>
	<p>Блок вычисления тригонометрических функций Trigonometric Function – выполняет вычисление некоторых тригонометрических функций, которые можно выбрать из списка.</p>
	<p>Блок вычисления действительной и мнимой части комплексного числа Complex to Real-Imag и блок, производящий обратную операцию Real-Imag to Complex.</p>
	<p>Блок вычисления модуля и аргумента комплексного числа Complex to Magnitude-Angle и блок, производящий обратную операцию Magnitude-Angle to Complex.</p>
	<p>Блок определения минимального или максимального значения MinMax – определяет максимальное или минимальное значение из всех сигналов, поступающих на его входы.</p>
	<p>Блок задания степенного многочлена Polynomial – задает степенной многочлен с помощью вектора его коэффициентов. Коэффициенты расположены в векторе по убыванию степени независимой переменной. Например, для полинома x^2+2x+5 необходимо задать вектор коэффициентов [1 2 5]. Коэффициенты должны быть действительного типа.</p>
	<p>Блок алгебраического контура Algebraic Constraint – выполняет поиск корней алгебраических уравнений. Задается начальное приближение выходного сигнала. Блок находит такое значение выходного сигнала, при котором значение входного сигнала становится равным нулю. При этом входной сигнал должен быть прямо или опосредованно связан с входным сигналом.</p>
	<p>Блок округления числового значения Rounding Function – выполняет операцию округления. Способ округления выбирается из списка:</p> <ul style="list-style-type: none"> • floor – до ближайшего меньшего целого,


	<ul style="list-style-type: none"> • ceil – до ближайшего большего целого, • round – до ближайшего целого, • fix – отбрасыванием дробной части. <p>Выходной сигнал блока будет иметь тип double или single.</p>
	<p>Блок определения знака сигнала Sign – работает в соответствии со следующим алгоритмом:</p> <ul style="list-style-type: none"> • Если входной сигнал положителен, то выходной сигнал равен 1. • Если входной сигнал отрицателен, то выходной сигнал равен -1. • Если входной сигнал равен 0, то выходной сигнал также равен 0.
	<p>Ползунковый регулятор Slider Gain – обеспечивает изменение коэффициента усиления в процессе расчета. После двойного щелчка мышью по иконке блока, появляется окно с ползунком, который можно передвигать мышкой.</p>
	<p>Блок преобразования размерности сигнала Reshape – изменяет размерность векторного или матричного сигнала</p>






Часто реальные процессы в первом приближении могут быть описаны линейными дифференциальными уравнениями с постоянными коэффициентами. Такие модели могут включать в себя блоки библиотеки **Continuous** – блоки с автоматически определяемым шагом времени.

Continuous – блоки для описания линейных систем (шаг по времени определяется автоматически)	
	<p>Блок вычисления производной Derivative – выполняет численное дифференцирование входного сигнала. Данный блок используется для дифференцирования аналоговых сигналов. При дифференцировании дискретного сигнала с помощью блока Derivative его выходной сигнал будет представлять собой последовательность импульсов соответствующих моментам времени скачкообразного изменения дискретного сигнала.</p>
	<p>Интегрирующий блок Integrator – выполняет интегрирование входного сигнала. Блок включает начальные условия интегрирования.</p>
	<p>Блок модели динамического объекта State-Space - создает динамический объект, описываемый уравнениями в пространстве состояний:</p>

	$\dot{x} = Ax + Bu$ $y = Cx + Du$ <p>где x – вектор состояния, u – вектор входных воздействий, y – вектор выходных сигналов, A, B, C, D - матрицы: системы, входа, выхода и обхода, соответственно.</p>
 <p>Transfer Fcn</p>	<p>Блок передаточной функции Transfer Fcn – задает передаточную функцию в виде отношения полиномов. Она определяется как отношение изображения Лапласа выходной переменной к изображению Лапласа входной переменной при нулевых начальных условиях.</p>
 <p>Zero-Pole</p>	<p>Блок передаточной функции, заданный в виде нулей и полюсов Zero-Pole – задает передаточную функцию в виде отношения произведений.</p>
 <p>Transport Delay</p>	<p>Блок фиксированной задержки сигнала Transport Delay – При выполнении моделирования значение сигнала и соответствующее ему модельное время сохраняются во внутреннем буфере блока Transport Delay. По истечении времени задержки значение сигнала, извлекается из буфера и передается на выход блока. В том случае, если шаги модельного времени не совпадают со значениями моментов времени для записанного в буфер сигнала, блок Transport Delay выполняет аппроксимацию выходного сигнала. Если начального значения объема памяти буфера не хватит для хранения задержанного сигнала, Simulink автоматически выделит дополнительную память. После завершения моделирования в командном окне MATLAB появится сообщение с указанием нужного размера буфера</p>
 <p>Variable Transport Delay</p>	<p>Блок управляемой задержки сигнала Variable Transport Delay – Выполняет задержку входного сигнала, заданную величиной сигнала управления.</p>

Блоки, описывающие нелинейные системы, собраны в библиотеке **Discontinuities**.

Discontinuities – блоки	
 <p>Backlash</p>	<p>Блок люфта Backlash – позволяет моделировать нелинейность типа «мертвый ход». Сигнал на выходе будет равен заданному значению Initial output, пока</p>

	<p>входной сигнал при возрастании не достигнет значения (Deadband width)/2 (где U – входной сигнал), после чего выходной сигнал будет равен U - (Deadband width)/2. После того как, произойдет смена направления изменения входного сигнала, он будет оставаться неизменным, пока входной сигнал не изменится на величину (Deadband width)/2, после чего выходной сигнал будет равен U+(Deadband width)/2</p>
 Saturation	<p>Блок ограничения Saturation – выполняет ограничение величины сигнала. Можно задавать верхний и нижний пределы для величины сигнала.</p>
 Coulomb & Viscous Friction	<p>Блок сухого и вязкого трения Coulomb and Viscous Friction – Блок реализует нелинейную характеристику, соответствующую выражению:</p> $y = \text{sign}(u) * (\text{Gain} * \text{abs}(u) + \text{Offset}),$ <p>где u – входной сигнал, y – выходной сигнал, Gain – коэффициент вязкого трения , Offset – величина сухого трения.</p>
 Dead Zone	<p>Блок с зоной нечувствительности Dead Zone – Если величина входного сигнала находится в пределах зоны нечувствительности, то выходной сигнал блока равен нулю. Если входной сигнал больше или равен верхнему входному порогу зоны нечувствительности, то выходной сигнал равен входному минус величина порога. А если входной сигнал меньше или равен нижнему порогу зоны нечувствительности, то выходной сигнал равен входному минус величина порога</p>
 Relay	<p>Релейный блок Relay – Выходной сигнал блока может принимать два значения. Одно из них соответствует включенному состоянию реле, второе – выключенному. Переход из одного состояния в другое происходит скачком при достижении входным сигналом порога включения или выключения реле. В том случае если пороги включения и выключения реле имеют разные значения, то блок реализует релейную характеристику с гистерезисом. При этом значение порога включения должно быть больше, чем значение порога выключения.</p>
 Rate Limiter	<p>Блок ограничения скорости изменения сигнала Rate Limiter – обеспечивает ограничение скорости изменения сигнала (первой производной).</p>



Блок квантования по уровню Quantizer – Блок обеспечивает квантование входного сигнала с одинаковым шагом по уровню.

В основной библиотеке **Simulink** выделены две группы блоков **Continuous** и **Discrete**. Блоки из той и другой группы предназначены для моделирования линейных систем, но они отличаются методом обработки информации (сигналов). Для выполнения вычислений в **Simulink** используются различные численные методы, но в любом случае сигнал представляется в виде последовательности отсчётов. Принципиальная разница между блоками из группы **Continuous** и группы **Discrete** заключается в выборе шага по времени.


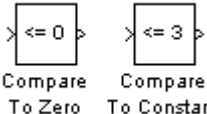


Блоки из группы **Continuous** предназначены для моделирования линейных систем, в которых сигнал имеет непрерывную зависимость от времени. Под непрерывной зависимостью в данном случае понимается то, что значение сигнала может быть найдено для любого значения времени. В процессе моделирования сигнал всё равно представляется последовательностью отсчётов, но моменты времени, которым соответствуют эти отсчёты, выбираются исходя из ограничений шага по времени и точности в настройках решателя.


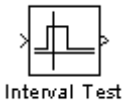
Блоки из группы **Discrete** предназначены для моделирования дискретных линейных систем. В процессе моделирования в этих блоках доступно только значение сигнала через определённые значения времени, определяемые периодом квантования. Поэтому, каждый блок из группы **Discrete** имеет параметр **Sampling time** (период квантования или величина шага дискретизации по времени). Есть возможность задать эту величину как вектор размерности 1×2 , тогда первым компонентом указывается период квантования, а вторым компонентом – значение времени смещения. Например, если блок имеет период квантования равный 1,5 с и при этом время смещения не задано, то в этом случае выборка выходных переменных производится в моменты времени квантования $t = 0; 1,5; 3,0; \dots$ с. Если период квантования и время смещения заданы как $[1.5 \ 0,5]$, то моменты времени квантования будут следующими $t = 0,5; 2,0; 3,5; \dots$ с. Если необходимо задать период квантования такой же, как и у предыдущего присоединённого блока, то вместо явного задания параметра в поле может быть указано -1. Это эффективно, так как при изменении периода квантования первого блока, этот параметр автоматически изменится во всех остальных присоединённых блоках.

Библиотека **Discrete** содержит блоки численного интегрирования и дифференцирования, передаточной функции и др. для дискретных систем. Перечислим блоки, которые могут понадобиться в моделях, где используются «непрерывные» блоки.

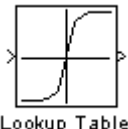
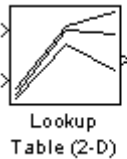
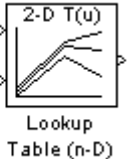
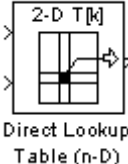
Discrete – элементы дискретных систем	
 Unit Delay	Блок единичной дискретной задержки Unit Delay – Выполняет задержку входного сигнала на один шаг модельного времени Sample time .
 Zero-Order Hold	Блок экстраполятора нулевого порядка Zero-Order Hold – Блок выполняет дискретизацию входного сигнала по времени с величиной шага дискретизации по времени Sample time . Блок фиксирует значение входного сигнала в начале интервала квантования и поддерживает на выходе это значение до окончания интервала квантования. Затем выходной сигнал изменяется скачком до величины входного сигнала на следующем шаге квантования.
 First-Order Hold	Блок экстраполятора первого порядка First-Order Hold – Блок задает линейное изменение выходного сигнала на каждом такте дискретизации, в соответствии с крутизной входного сигнала на предыдущем интервале дискретизации.
 Memory	Блок Memory – выполняет задержку входного сигнала на один временной такт.

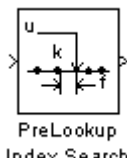
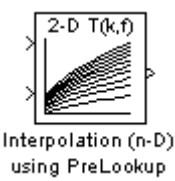
Блоки библиотеки **Logic and Bit Operations** выполняют логические операции. Выходным сигналом этих блоков является **1**, если результат вычисления операции есть “**ИСТИНА**” и **0**, если результат – “**ЛОЖЬ**”. На вкладке **Signal data types** окна параметров блока можно выбрать формат выводимого сигнала. По умолчанию это логический тип (**boolean**). Но также можно выбрать некоторые числовые форматы.

Logic and Bit Operations – логические и побитовые операции	
 Relational Operator	Блок вычисления операции отношения Relational Operator –сравнивает текущие значения входных сигналов. Тип операции отношения можно выбрать из списка.
 Compare To Zero Compare To Constant	Блок вычисления операции сравнения с нулем или с константой. Аналогичен блоку Relational Operator .
 Logical Operator	Блок логических операций Logical Operation – реализует одну из базовых логических операций, которую можно выбрать из списка.
 Bitwise Operator	Блок побитовых логических операций Bitwise Operator – реализует одну из базовых логических операций по отношению к целому числу в двоичном представлении. Одним из операндов блока является


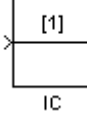
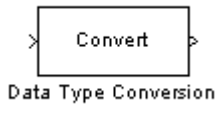
	сигнал, подаваемый на вход блока, а вторым – параметр блока Second operand . Второй операнд. Задается шестнадцатеричным числом в символьном виде. Входными сигналами блока должны быть беззнаковые переменные типа uint8 , uint16 или uint32 .
 <p>Combinatorial Logic</p>	Блок комбинаторной логики Gombinatorical Logic – преобразует входные сигналы в соответствии с таблицей истинности Truth table .
 <p>Interval Test</p>	Блок выбора интервала Interval Test – выходной сигнал этого блока 1, если входной сигнал лежит в заданных пределах, и 0, если входной сигнал выходит за границы.

Перечислим некоторые блоки, позволяющие работать с таблицами данных.


Lookup Tables – таблицы	
 <p>Lookup Table</p>	Блок одномерной таблицы Look-Up Table – задает в табличной форме функцию одной переменной. Таблицу можно задавать в виде дискретных значений входного и выходного сигнала (например, вектор входных значений [0 1 2 5] , а вектор выходных значений [-5 -10 3 100]) или в виде непрерывного диапазона или вычисляемого выражения (например, вектор входных значений равен [-5:5] , а вектор выходных значений равен tanh([-5:5])). Если входной сигнал не совпадает ни с одним из элементов вектора входных значений, то блок выполняет линейную интерполяцию между двумя ближайшими к нему элементами или линейную экстраполяцию по двум крайним элементам.
 <p>Lookup Table (2-D)</p>  <p>Lookup Table (n-D)</p>	Блок двумерной (n-мерной) таблицы Look-Up Table(2D) – задает в табличной форме функцию двух переменных. Таблица значений функции задается в виде матрицы, заданная пользователем нумерация строк которой соответствует вектору значений первого элемента, а колонок – второго аргумента.
 <p>Direct Lookup Table (n-D)</p>	Блок таблицы с прямым доступом Direct Loop-Up Table (n-D) – задает многомерную таблицу с прямым доступом к ее элементам. Индексация элементов начинается с нуля. На выходе блока можно получить элемент таблицы, столбец или матрицу. Для этого на вход надо подавать значения всех индексов элемента, на один индекс меньше или на два индекса меньше.

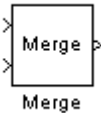
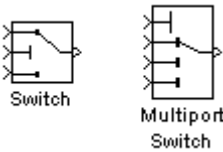
	ше соответственно.
	Блок работы с индексами PreLook-Up Index Search – вычисляет значение индекса и относительную величину входного сигнала. Используется совместно с блоком Interpolation (n-D) using PreLook-Up .
	Блок интерполяции табличной функции Interpolation (n-D) using PreLook-Up – вычисляет значение табличной функции по значению индекса и относительной величине входного сигнала. Используется совместно с блоками PreLook-Up Index Search .

Signal Attributes – характеристики сигналов

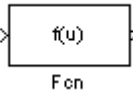
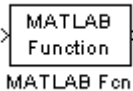
	Блок определения размерности сигнала Width – вычисляет размерность входного сигнала.
	Блок установки начального значения сигнала IC – задает начальное значение сигнала Initial value . Выходной сигнал блока равен значению параметра Initial value на первом шаге расчета вне зависимости от величины входного сигнала блока. На остальных расчетных шагах входной сигнал проходит на выход блока без каких-либо изменений.
	Блок преобразования типа сигнала Data Type Conversion – Тип данных выходного сигнала может принимать значения (выбираются из списка): auto , double , single , int8 , int16 , int32 , uint8 , uint16 , uint32 и boolean .

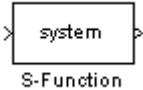
Signal Routing – маршрутизация сигналов

	Мультиплексор (смеситель) Mux – объединяет входные сигналы в вектор и блок обратной операции Демльтиплексор (разделитель) Demux – разделяет входной векторный сигнал на отдельные составляющие.
	Блок селектора Selector – выбирает из вектора или матрицы требуемые элементы.
	Блок объединения сигналов в матрицу Matrix Concatenation – выполняет объединение (конкатенацию) входных векторов или матриц.
	Блок передачи сигнала Goto – выполняет передачу сигнала к блоку приема сигнала From без линии связи.


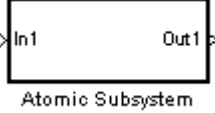
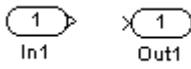
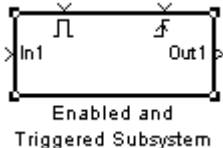
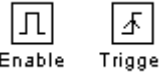
	<p>Блок объединения сигналов Merge – выполняет объединение входных сигналов в единый векторный сигнал.</p>
	<p>Блок переключателя Switch – выполняет переключение входных сигналов по сигналу управления. Блок работает следующим образом: если сигнал управления, подаваемый на средний вход больше, чем величина порогового значения Threshold, то на выход блока проходит сигнал с первого (верхнего) входа. Если сигнал управления станет меньше, чем пороговое значение, то на выход блока будет поступать сигнал со второго (нижнего) входа. Аналогично, блок многовходового переключателя Multiport Switch выполняет переключение входных сигналов по сигналу управления, задающему номер активного входного порта.</p>

Блоки из библиотеки **User-Defined Functions** (функции, задаваемые пользователем) позволяют совмещать возможности **Simulink** и **MATLAB** и самостоятельно писать новые **Simulink**-функции.

User-Defined Functions – функции, задаваемые пользователем	
	<p>Блок задания функции Fcn – В окне Expression можно задать математическое выражение (по правилам, принятым для описания функций на языке C). Входной сигнал в выражении обозначается u (если он является скаляром). Если входной сигнал – вектор, необходимо указывать номер элемента вектора в круглых скобках. Например, u(1) и u(3) – первый и третий элементы входного вектора. Также можно использовать арифметические и логические операторы, операторы отношения, математические функции, переменные из рабочей области. Блок не поддерживает матричные и векторные операции. Выходной сигнал блока всегда – скаляр.</p>
	<p>Блок задания функции MATLAB Fcn – задает выражение в стиле языка программирования MATLAB. Входной сигнал в окне MATLAB function обозначается u (скаляр) или u(i) – i-ый элемент вектора. Задаваемое выражение может содержать собственные функции пользователя, написанные на языке MATLAB и оформленные в виде m-файлов. Имя m-файла не должно совпадать с именем модели (mdl-файлом).</p>

	<p>Блок S-function – подключает к модели Simulink-функцию. S-функция является описанием блока на одном из языков программирования: MATLAB, C, C++, Ada, или Fortran. Имя S-функции не должно совпадать с именем модели (mdl-файла).</p>
---	--

Блоки библиотеки **Ports&Subsystems** – порты и подсистемы используются для создания различных подсистем в **Simulink**.

Ports&Subsystems – порты и подсистемы	
	<p>Виртуальная подсистема Subsystem.</p>
	<p>Монолитная подсистема Atomic Subsystem.</p>
	<p>Входной In и выходной Out порты подсистемы, осуществляющие связь подсистемы с основной моделью.</p>
	<p>Управляемая уровнем и фронтом сигнала подсистема Enabled and Triggered Subsystem.</p>
	<p>Управляющие элементы Enable и Trigger, находящиеся в окне управляемой подсистемы и изменяющие ее свойства управления.</p>

2.5. Подсистемы в Simulink

Подсистема – это фрагмент **Simulink**-модели, оформленный в виде отдельного блока. Использование подсистем при составлении модели имеет следующие преимущества:

1. уменьшает количество одновременно отображаемых блоков на экране, что облегчает восприятие модели (в идеале модель полностью должна отображаться на экране монитора);
2. позволяет создавать и отлаживать фрагменты модели по отдельности, что повышает технологичность создания модели;
3. позволяет создавать собственные библиотеки;
4. дает возможность синхронизации параллельно работающих подсистем;
5. позволяет включать в модель собственные справочные средства;
6. дает возможность связывать подсистему с каким-либо **m-файлом**, обеспечивая запуск этого файла при открытии подсистемы (нестандартное открытие подсистемы).

Использование подсистем позволяет создавать блоки, не уступающие стандартным по своему оформлению (собственное окно параметров блока, пиктограмма, справка и т.п.). Количество подсистем в модели не ограничено, кроме того подсистемы могут включать в себя другие подсистемы. Уровень вложенности подсистем друг в друга также не ограничен.

Связь подсистемы с моделью (или с подсистемой верхнего уровня иерархии) выполняется с помощью входных (блок **In**) и выходных (блок **Out**) портов. Добавление в подсистему входного или выходного порта приводит к появлению на изображении подсистемы метки порта, с помощью которой внешние сигналы передаются внутрь подсистемы или выводятся в основную модель. Переименование блоков **In** или **Out** позволяет изменить метки портов, отображаемые на пиктограмме подсистемы, со стандартных (**In** и **Out**) на те, которые нужны пользователю.

Подсистемы могут быть виртуальными (**Subsystem**) и монолитными (**Atomic Subsystem**). Отличие этих видов подсистем заключается в порядке выполнения блоков во время расчета. Если подсистема является виртуальной, то **Simulink** игнорирует наличие границ отделяющих такую подсистему от модели при определении порядка расчета блоков. Иными словами в виртуальной системе сначала могут быть рассчитаны выходные сигналы нескольких блоков, затем выполнен расчет блоков в основной модели, а затем вновь выполнен расчет блоков входящих в подсистему. Монолитная подсистема считается единым (неделимым) блоком и **Simulink** выполняет расчет всех блоков в такой подсистеме, не переключаясь на расчеты других блоков в основной модели. Изображение монолитной подсистемы имеет более толстую рамку по сравнению с виртуальной подсистемой.

Подсистемы могут быть также управляемыми или неуправляемыми. Управляемые подсистемы всегда являются монолитными. Управляемые подсистемы имеют дополнительные (управляющие) входы, на которые поступают сигналы, активизирующие данную подсистему. Управляющие входы расположены сверху или снизу подсистемы. Когда управляемая подсистема активизирована – она выполняет вычисления. В том случае если управляемая подсистема пассивна, то она не выполняет вычисления, а значения сигналов на ее выходах определяются настройками выходных портов.

Для создания в модели подсистемы можно воспользоваться двумя способами:

Первый способ: перенести блок **Subsystem** (или другой вид подсистемы) из библиотеки **Ports&Subsystems** в окно основной модели. Двойным щелчком мыши по блоку подсистемы открыть рабочее окно подсистемы и создавать модель как обычно. Обмен данными с основной моделью будет производиться с помощью блоков **In** или **Out**. На Рис. 55 показаны окно основной модели с блоком **Subsystem** и окно подсистемы с входным и выходным блоками.

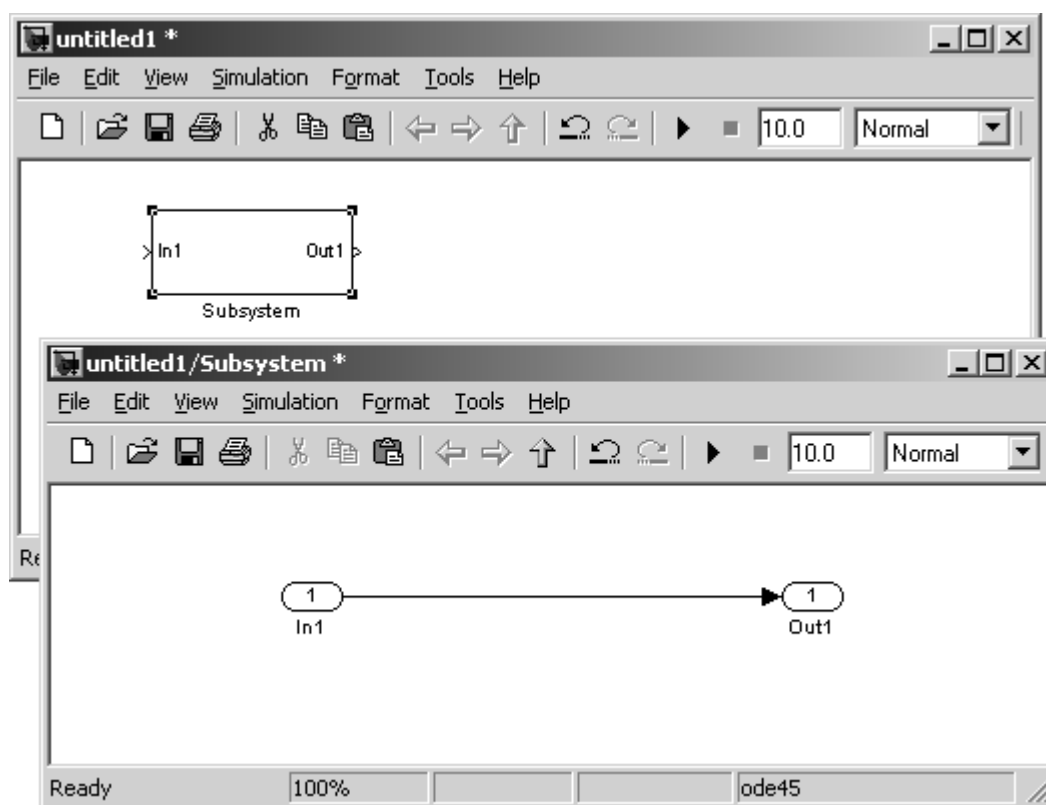


Рис. 55. Окна основной модели и подсистемы

Сначала входной и выходной блоки соединены. Можно удалить эту связь и строить в этом окне нужную схему так, чтобы все входящие в подсистему связи были соединены с портами **In**, а все выходящие связи с – **Out**. Порты **In** и **Out** можно при необходимости добавлять из библиотеки.

Рассмотрим пример простой схемы с подсистемой. На Рис. 56 показаны окна основной модели и подсистемы. На вход подсистемы подается повторяющаяся последовательность линейно возрастающего сигнала. Подсистема состоит всего из одного блока «Интегратор» (не считая портов входа и выхода) и производит интегрирование этого сигнала, исходный сигнал и результат интегрирования регистрируются осциллографом.

Результат работы этой схемы показан на Рис. 57. Исходная последовательность треугольных импульсов изображена сплошной линией, а интеграл – пунктирной. На графике видно, что в параметрах «Интегратора» начальное условие интегрирования выбрано -2.

Второй способ: выделить с помощью мыши нужный фрагмент модели, включая все блоки и линии связи будущей подсистемы, и выполнить команду **Create Subsystem** из меню **Edit** окна модели. Выделенный фрагмент будет помещен в подсистему, а входы и выходы подсистемы будут снабжены соответствующими портами. Данный способ позволяет создать виртуальную неуправляемую подсистему. В дальнейшем, если это необходимо, можно сделать подсистему монолитной, изменив ее параметры, или управляемой, добавив управляющий элемент из библиотеки. Отменить группировку блоков в подсистему можно командой **Undo**.

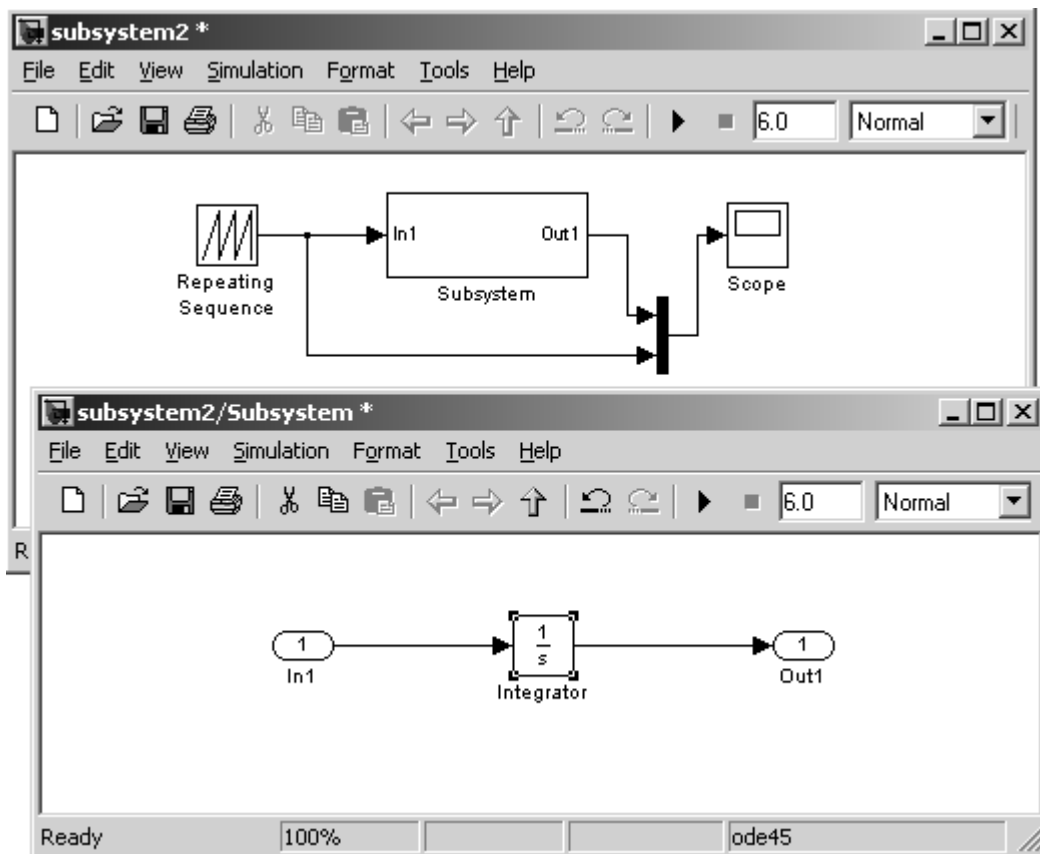


Рис. 56. Пример подсистемы

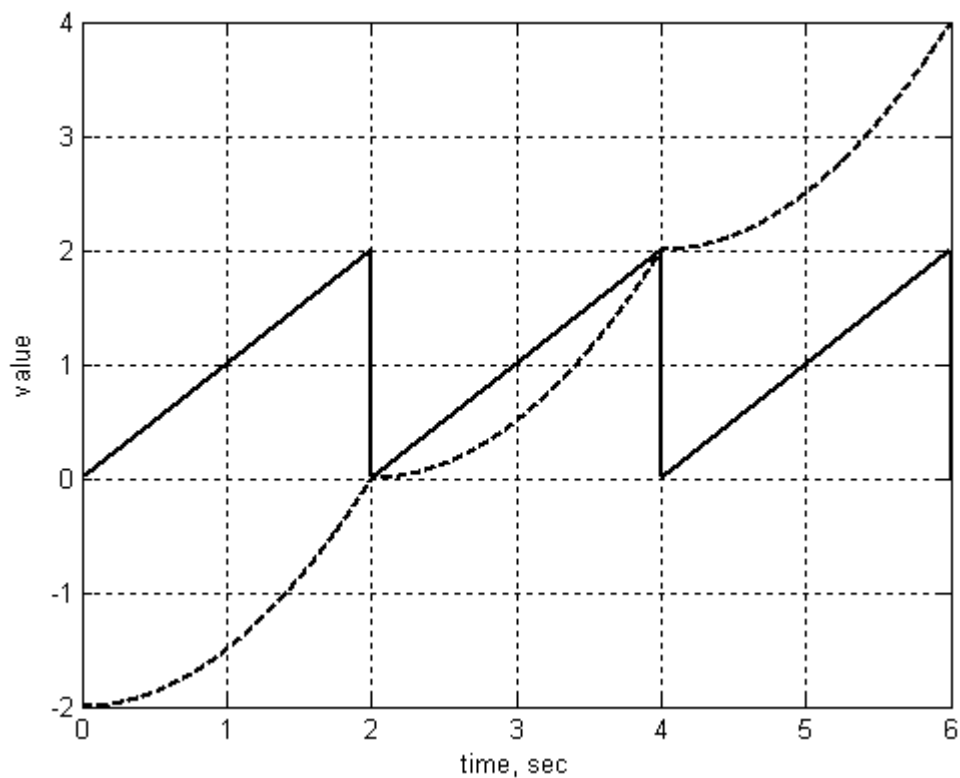


Рис. 57. Интегрирование последовательности треугольных импульсов

Рассмотрим этот способ на примере рассмотренной выше модели колебательного контура. На Рис. 58 пунктирной линией выделена часть модели, которая войдет в подсистему.

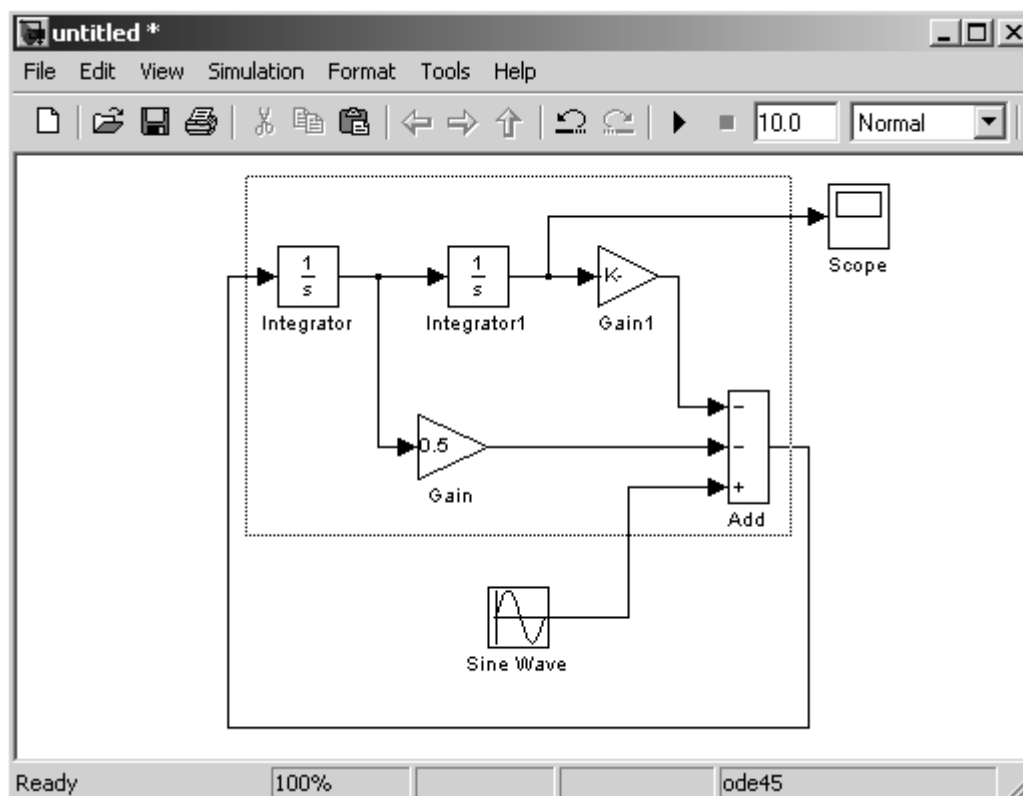


Рис. 58. Создание подсистемы: выделение

После выполнения команды **Edit \Create Subsystem** выделенная часть модели преобразуется в подсистему. Основное окно модели будет выглядеть, как показано на Рис. 59.

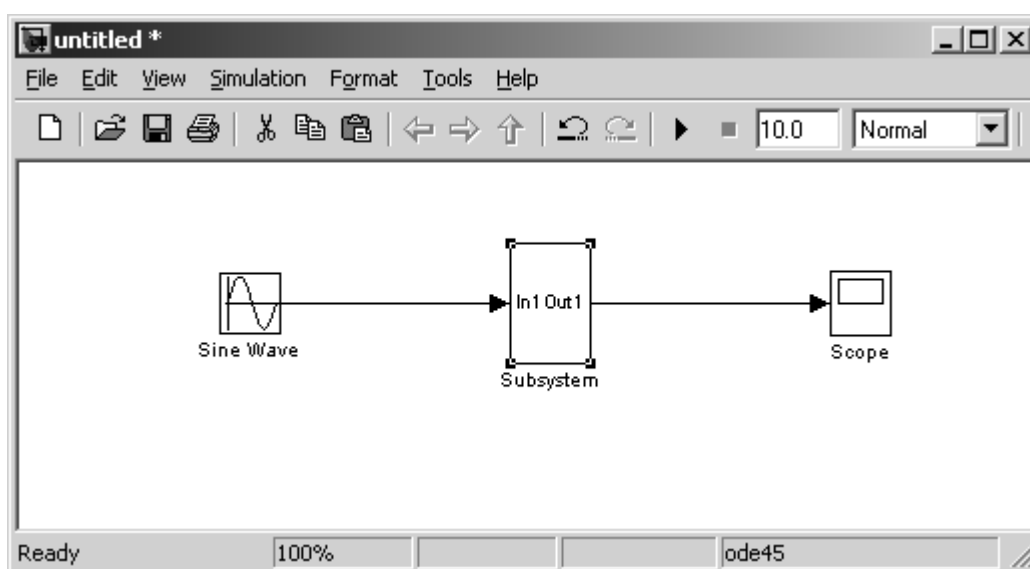


Рис. 59. Создание подсистемы: основное окно модели

Двойной щелчок мыши по блоку **Subsystem** откроет рабочее окно подсистемы (Рис. 60). Здесь к основным блокам подсистемы добавлены блоки **In** для входной переменной и блок **Out** для выходной переменной подсистемы. Изменение имен этих блоков приведет к соответствующему изменению надписей на изображении блока всей подсистемы.

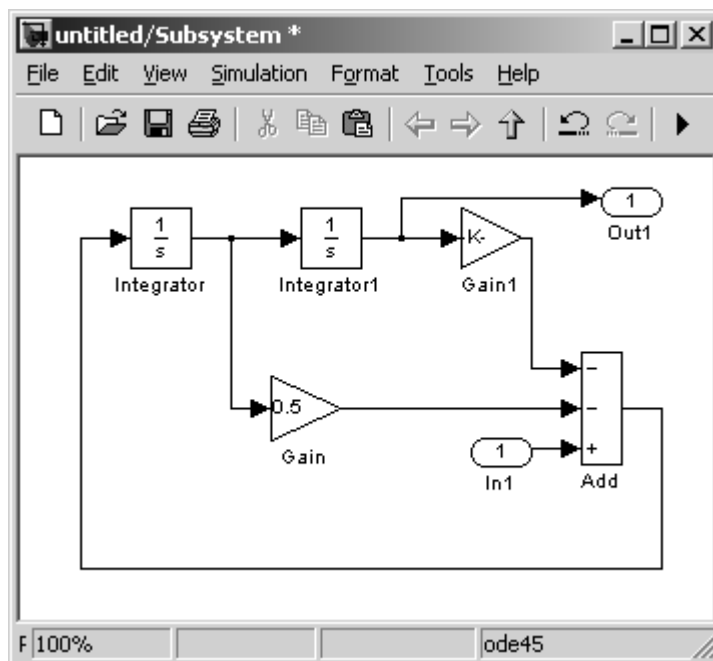


Рис. 60. Рабочее окно подсистемы

Можно изменить размер блока **Subsystem** так, чтобы имена портов были читаемы и произвести перестановку блоков для улучшения внешнего вида схемы.

Команда **Edit \Create Subsystem** не имеет обратной операции. Как только группа блоков объединена в подсистему, отменить данную операцию уже невозможно. Если требуется извлечь структурную схему из подсистемы, то для этого нужно скопировать подсистему в новое окно (удалив из основного окна), открыть ее структуру и затем скопировать блоки из окна подсистемы в окно основной модели.

Кратко перечислим основные параметры разных видов подсистем.

Виртуальная и монолитная подсистемы **Subsystem** и **Atomic Subsystem**:

Доступ к окну параметров подсистемы осуществляется через меню **Edit** окна основной модели командой **SubSystem Parameters...** Окно параметров изображено на Рис. 61. Перечислим параметры этого окна:

1. **Show port labels** – показать метки портов.
2. **Treat as atomic unit** (флажок) – считать подсистему монолитной. Таким образом, блоки виртуальной и монолитной подсистем – это один и тот же блок, отличающийся значением данного параметра.

3. **Read/Write permissions** – доступность подсистемы для изменений. Выбирается из списка:

- **ReadWrite** – пользователь может открывать и изменять подсистему.
- **ReadOnly** – пользователь может открывать подсистему только для просмотра.
- **NoReadOrWrite** – пользователь не может открывать и изменять подсистему.

4. **Name of error callback function** – имя функции, используемой для обработки ошибок, возникающих в данной подсистеме.

Остальные параметры подсистемы доступны при разработке приложений с использованием **Real-Time Workshop** и рассмотрены в документации на это приложение.

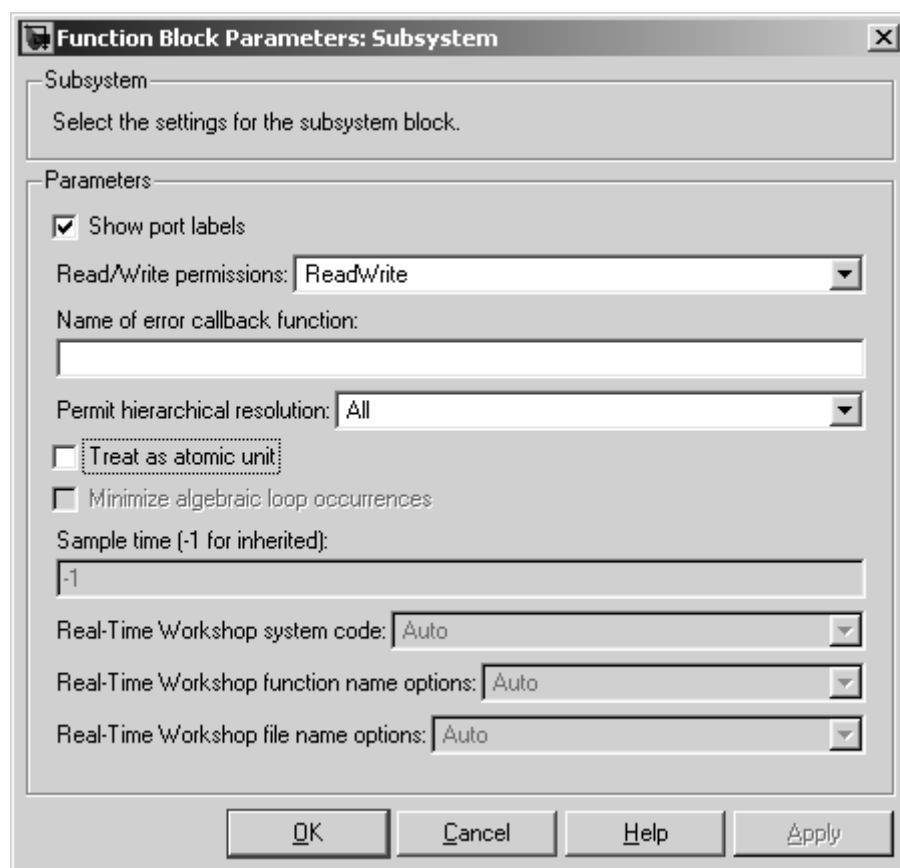


Рис. 61. Окно параметров подсистемы

Управляемая уровнем сигнала подсистема **Enabled Subsystem**.

Подсистема **Enabled Subsystem** (в дальнейшем **Е**-подсистема) активизируется при наличии положительного сигнала на управляющем входе. Если входной сигнал векторный, то подсистема активизируется, если хотя бы один элемент принимает положительное значение. Величина выходного сигнала в том случае, если система заблокирована, определяется настройками выходных портов подсистемы (блоки **Outputport**). В том случае если параметр **Output when disabled** (вид сигнала на выходе подсистемы) выходного порта имеет

значение **held**, то выходной сигнал подсистемы равен последнему рассчитанному ею значению, если же этот параметр имеет значение **reset**, то выходной сигнал подсистемы равен значению, задаваемому параметром **Initial output** (начальное значение).

Свойства Е-подсистемы определяются параметрами блока **Enable** (Рис. 62), который может находиться в любом месте данной подсистемы.



Рис. 62. Пиктограмма блока Enable

Этот блок первоначально находится в окне блока **Enabled Subsystem** или его можно перетащить в окно виртуальной подсистемы **Subsystem** из библиотеки блоков. Окно параметров блока **Enable** показано на Рис. 63.

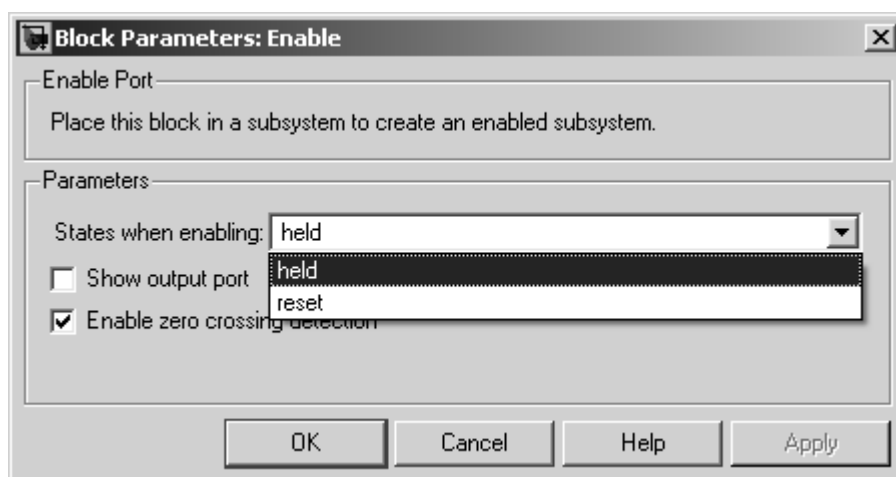


Рис. 63. Окно параметров блока Enable

1. **States when enabling** – состояние при запуске. Параметр задает состояние подсистемы при каждом запуске. Выбирается из списка:
 - **held** – Использовать предыдущее состояние (последнее состояние, когда система была активна).
 - **reset** – Использовать начальное (исходное) состояние.
2. **Show output port** (флажок) – показать выходной порт. При установленном флажке на пиктограмме блока **Enable** появляется дополнительный выходной порт, сигнал с которого может быть использован для управления блоками внутри подсистемы.

Управляемая фронтом сигнала подсистема **Triggered Subsystem**.

Подсистема **Triggered Subsystem** (в дальнейшем **Т-подсистема**) включается фронтом (перепадом уровня) управляющего сигнала и выполняет вычисления только на том шаге моделирования, где произошло это изменение.

Если входной сигнал векторный, то подсистема активизируется, если хотя бы в одном элементе изменяется уровень сигнала. Возврат Т-подсистемы в исходное состояние не производится (подсистема сохраняет последнее значение до следующего запуска), поэтому параметр **States when enabling** выходных портов имеет значение **held**, и недоступен для изменения.

В Т-подсистеме могут использоваться блоки, для которых модельное время является наследуемым параметром от предыдущего блока (например, **Gain** или **Logical Operator**), а также дискретные блоки, для которых параметр **sample time** имеет значение **-1** (минус один).

Свойства Т-подсистемы определяются параметрами блока **Trigger**, который может находиться в любом месте данной подсистемы. Его, также как и блок **Enable** можно добавлять в окно виртуальной подсистемы. Пиктограмма блока **Trigger** показана на Рис. 64.



Рис. 64. Пиктограмма блока Trigger

Окно параметров этого блока показано на Рис. 65.

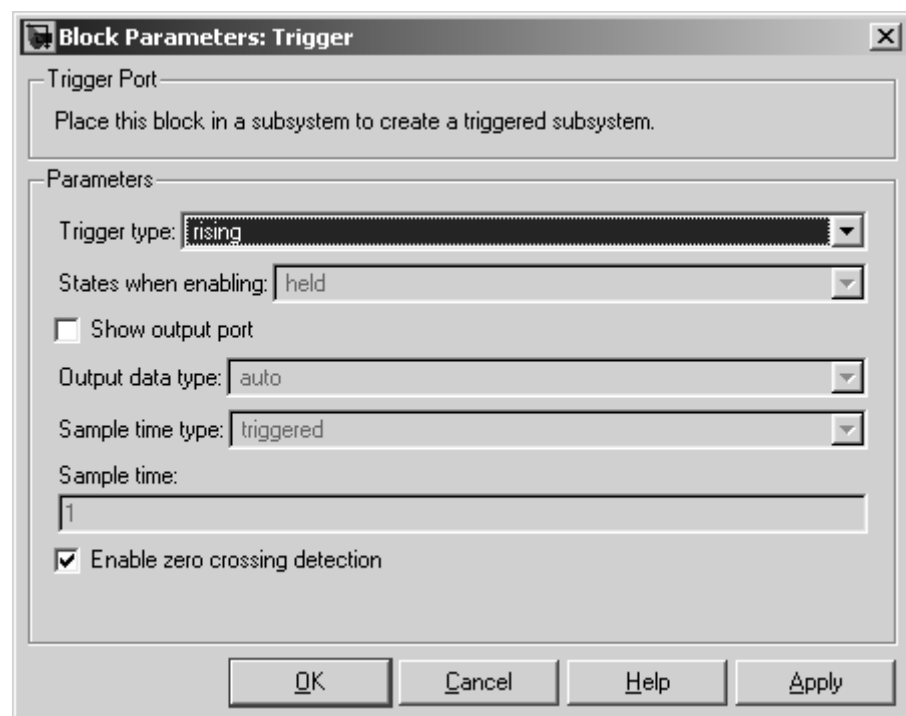


Рис. 65. Окно параметров блока Trigger

1. **Trigger type** – тип триггера. Выбирается из списка:
 - **rising** – активизация подсистемы положительным фронтом.
 - **falling** – активизация подсистемы отрицательным фронтом.

- **either** – активизация подсистемы как положительным, так и отрицательным фронтом.
 - **function-call** – активизация подсистемы определяется логикой работы заданной S-функции.
2. **Show output port** (флажок) – показать выходной порт.

Управляемая уровнем и фронтом сигнала подсистема **Enabled and Triggered Subsystem** .

Подсистема **Enabled and Triggered Subsystem** (в дальнейшем **ЕТ**-подсистема) включается фронтом сигнала, поступающего на **Т**-вход системы, при наличии положительного сигнала на **Е**-входе системы. Так же как и **Triggered Subsystem**, эта подсистема выполняет вычисления только на том шаге моделирования, где произошло изменение управляющего сигнала на **Т**-входе. Параметр **States when enabling** блока **Enable** не оказывает влияния на работу **ЕТ**-подсистемы.

Оба управляющих сигнала могут быть векторными.

2.6. Маскирование подсистем в Simulink

Механизм маскирования подсистем позволяет оформить подсистему как полноценный библиотечный блок, т.е. снабдить подсистему собственным окном параметров, пиктограммой, справочной системой и т.п.

Маскирование подсистем дает пользователю следующие преимущества:

1. Расширяет возможности пользователя по управлению параметрами модели.
2. Позволяет создавать более понятный интерфейс подсистемы.
3. Повышает наглядность блок-диаграммы.
4. Расширяет возможности построения сложных моделей.
5. Повышает защищенность модели от несанкционированной модификации.

Для выполнения маскирования имеющейся подсистемы необходимо предварительно выполнить следующие действия:

1. Определить, какие параметры подсистемы должны задаваться пользователем в будущем окне параметров. Задать эти параметры в подсистеме с помощью идентификаторов (имен).
2. Определить, каким образом параметр должен задаваться в окне диалога (с помощью строки ввода, выбором из раскрывающегося списка или установкой флажка).
3. Разработать эскиз пиктограммы блока.
4. Создать комментарии (справку) по использованию подсистемы.

Маскирование подсистемы выполняется с помощью **Mask Editor** (редактор маски). Для запуска редактора маски необходимо выделить маскируемую подсистему и выполнить команду **Mask Subsystem...** из меню **Edit**. Можно также воспользоваться контекстным меню. После запуска **Mask Editor** на экран будет выведено окно редактора (Рис. 66), имеющее 4 вкладки: **Icon** (Пиктограмма), **Parameters** (Параметры), **Initialization** (Инициализация), **Documentation** (Документация). Первая из вкладок обеспечивает создание пиктограммы подсистемы, вторая – задает параметры подсистемы, третья – дает возможность создать окно диалога для ввода параметров и четвертая – позволяет ввести описание блока и создать его справку.

Повторный вызов редактора маски для уже маскированной подсистемы осуществляется командой **Edit Mask...** из меню **Edit** (или аналогичной командой из контекстного меню).

После того, как маскирование системы будет выполнено, двойной щелчок на ее изображении будет открывать окно параметров подсистемы, а не окно модели. Открыть саму подсистему (окно модели) для редактирования или просмотра можно командой **Look under mask** из меню **Edit** или контекстного меню.

Рассмотрим вкладку **Icon** (Пиктограмма) редактора маски (Рис. 66). Эта вкладка содержит следующие элементы:

Основное окно **Drawing commands** – область ввода команд рисования. Команды рисования являются выражениями допустимыми в языке **MATLAB**. В нижней части окна находится выпадающий список с синтаксисом возможных команд (внизу приводится образец и справа – пиктограмма, показывающая результат выполнения этой команды).

Область слева с выпадающими списками **Icon options**:

1. **frame** – список позволяющий выбрать способ отображения рамки пиктограммы:

- **Visible** – рамка видна;
- **Invisible** – рамка не видна.

2. **Transparency** – список, позволяющий установить прозрачность пиктограммы:

- **Opaque** – пиктограмма не прозрачна;
- **Transparent** – пиктограмма прозрачна.

3. **Rotation** – список, позволяющий задать возможность вращения пиктограммы:

- **Fixed** – положение пиктограммы фиксировано;
- **Rotates** – пиктограмма может вращаться вместе с блоком.

4. **Units** – список, задающий условия масштабирования пиктограммы:

- **Autoscale** – автоматическое масштабирование. Рисунок занимает максимально возможную площадь внутри пиктограммы.

- **Normalized** – нормализованное масштабирование. Координаты левого нижнего угла пиктограммы **(0,0)**, координаты правого верхнего угла **(1,1)**.

- **Pixel** – координаты рисунка задаются в пикселях.

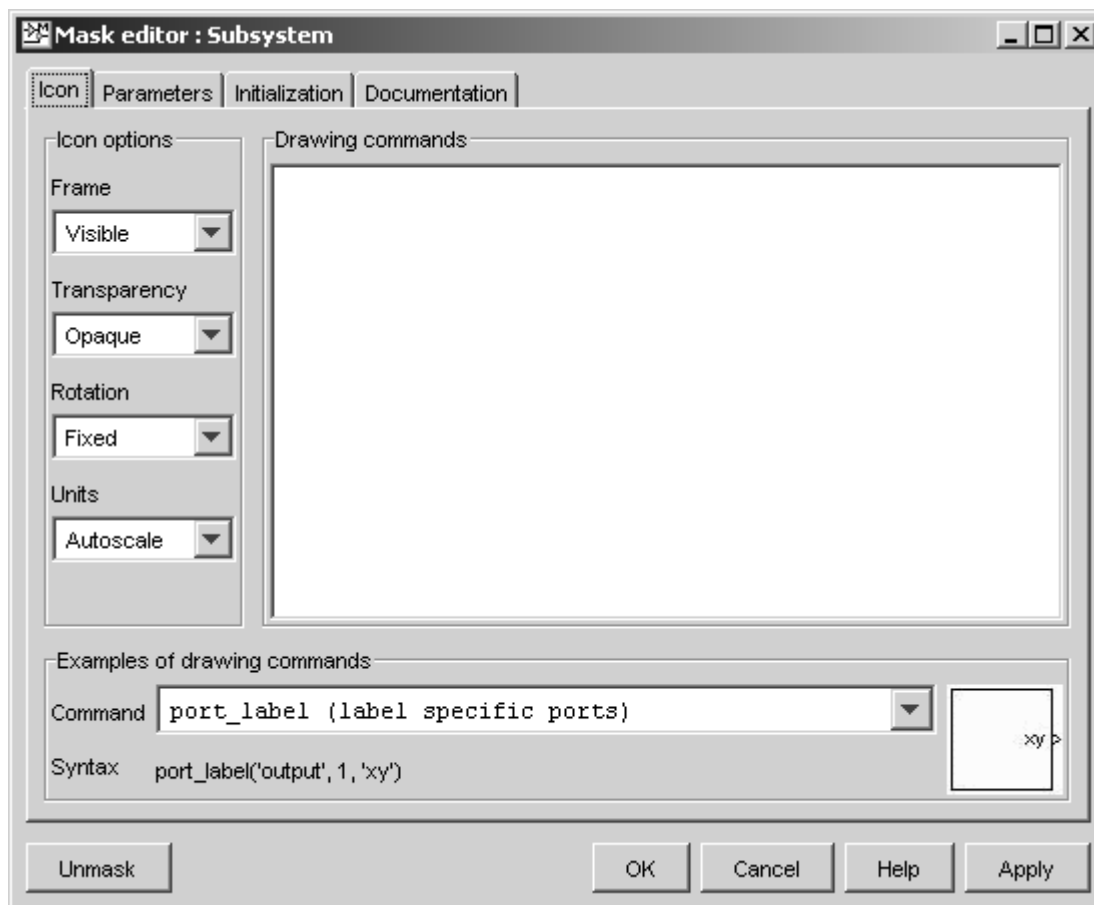


Рис. 66. Окно редактора маски: вкладка Icon

В нижней части окна имеется 5 кнопок управления редактором:

1. **OK** – сохранить внесенные изменения и закрыть окно.
2. **Cancel** – отменить внесенные изменения и закрыть окно.
3. **Unmask** – снять маску с подсистемы. До закрытия файла модели маску можно восстановить, воспользовавшись командой **Edit Mask...** из меню **Edit**.
4. **Help** – открыть окно справки редактора маски.
5. **Apply** – сохранить внесенные изменения без закрытия окна редактора.

Для вывода текста в окне **Drawing commands** могут использоваться следующие команды:

- **disp('text')** или **disp(variablename)** – вывод текста '**text**' или значения символьной переменной **variablename** в центре пиктограммы.
- **text(x, y, 'text')** или **text(x, y, variablename)** – вывод текста '**text**' или значения символьной переменной **variablename**, начиная с позиции, заданной координатами **x** и **y**.

- **text(x, y, 'text', 'horizontalAlignment', halign,'verticalAlignment', valign)** – вывод текста 'text' в позиции, заданной координатами x и y и с указанием способов выравнивания относительно этой позиции по вертикали или горизонтали. Параметр **halign** может принимать значения: 'left', 'right' или 'center'. Параметр **valign** может принимать значения: 'base', 'bottom' или 'middle'.

- **fprintf('text')** или **fprintf('format', variablename)** – форматированный вывод (по правилам языка C) текста 'text' или значения символьной переменной **variablename** в центре пиктограммы.

- **port_label(port_type, port_number, label)** – вывод на пиктограмме метки порта. Например, выражение **port_label('input', 1, 'a')** выводит на пиктограмме метку **a** первого входного порта.

Для вывода текста в несколько строк допускается использование сочетания символов **\n** для перехода на новую строку.

Примеры маскированных подсистем с различными вариантами текстовых надписей даны на Рис. 67. Значения текстовых переменных заданы на вкладке **Initialization** в графе **Initialization commands**, а сами команды – на вкладке **Icon** в графе **Drawing commands**.

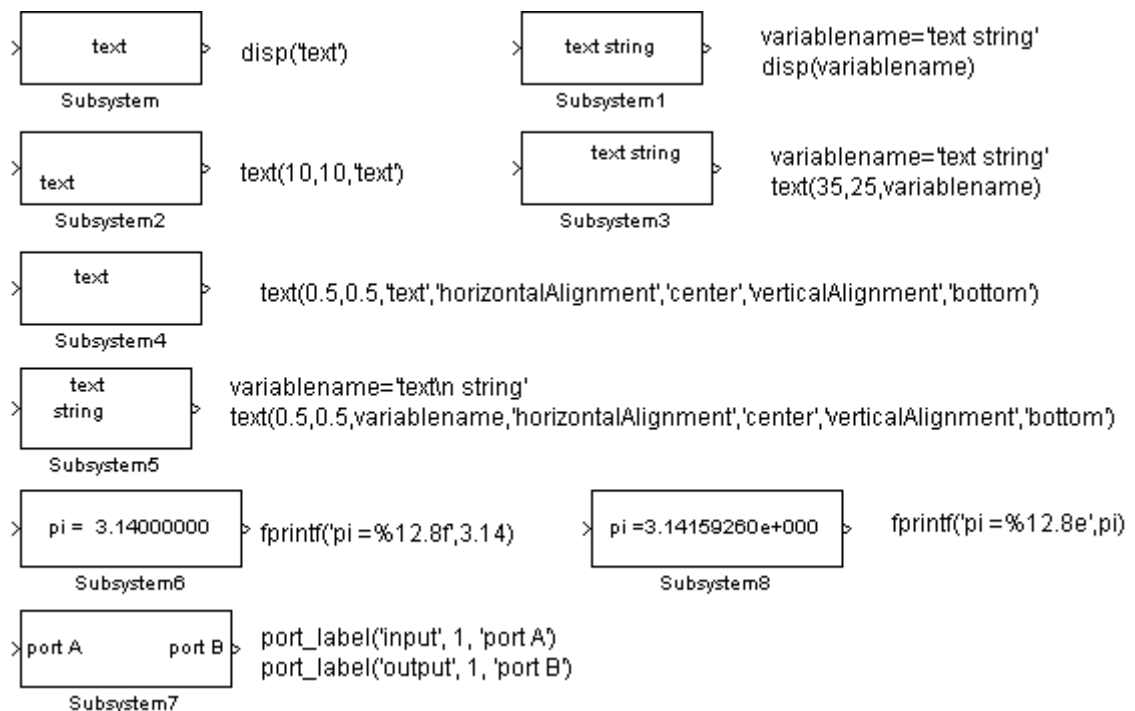


Рис. 67. Варианты текстовых надписей на пиктограммах

Для построения графиков на пиктограмме могут использоваться следующие команды:

- **plot(Y)** – в том случае, если **Y** является вектором, то строится график, по оси абсцисс которого откладывается значение индекса элемента, а по оси ординат значение самого элемента. В том случае если **Y** является матрицей, то строятся линии для каждого столбца. По оси абсцисс в этом случае также откладывается значение индекса элемента.
- **plot(X1,Y1,X2,Y2,...)** – строится графики вида **Y1(X1)**, **Y2(X2)** и т.д.

Примеры маскированных подсистем с различными вариантами графиков представлены на Рис. 68. Значения переменных задаются на вкладке **Initialization** в графе **Initialization commands**, а сами команды – на вкладке **Icon** в графе **Drawing commands**.

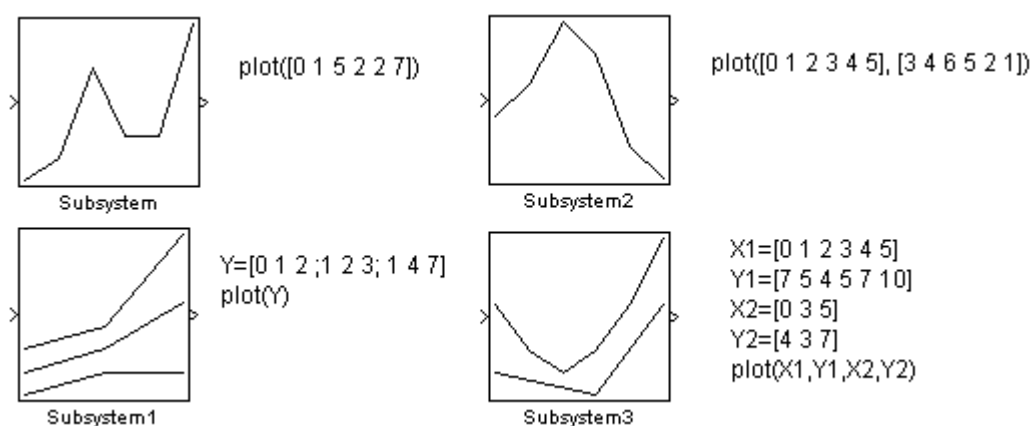


Рис. 68. Варианты графиков на пиктограммах

Для отображения на пиктограмме передаточной функции используются следующие команды:

- **dpoly(num, den)** – отображение дробно-рациональной передаточной функции (**num** – вектор коэффициентов числителя, **den** – вектор коэффициентов знаменателя). Оператор Лапласа будет отображен с помощью символа **s**.
- **dpoly(num, den, 'character')** - отображение дробно-рациональной передаточной функции. Оператор Лапласа будет отображен с помощью символа **character**.
- **dpoly(num, den, 'z')** – отображение дискретной дробно-рациональной передаточной функции.
- **dpoly(num, den, 'z-')** - отображение дискретной дробно-рациональной передаточной функции от обратного аргумента.
- **droots(z, p, k)** - отображение **Zpk**-формы передаточной функции. Для рассматриваемого выражения может быть добавлен четвертый аргумент в виде **'z'** или **'z-'** для отображения дискретных передаточных функций.

Примеры маскированных подсистем с различными вариантами отображения передаточных функций показаны на Рис. 69.

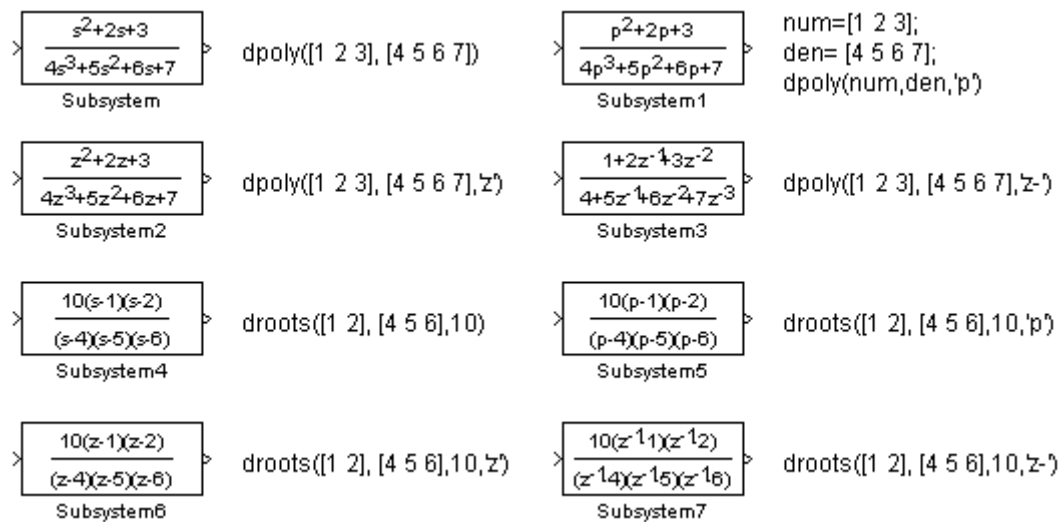


Рис. 69. Варианты отображения передаточных функций на пиктограммах

Для отображения на пиктограмме рисунка из графического файла используются следующие команды:

- **image(imread('filename'))** – отображение рисунка из файла с полным именем **filename**. Для правильной работы этой команды необходимо поместить рисунок в ту же папку, где находится файл модели, и сделать эту папку рабочей. Допускается также совместно с именем файла указывать его полный путь.

- **image(a, [x, y, w, h])** – отображение рисунка содержащегося в переменной **a**. Ширина и высота рисунка задаются параметрами **w** и **h**, соответственно. Левый нижний угол рисунка расположен в точке с координатами **x,y**. Считывание рисунка из файла может быть выполнено командой **a = imread('filename')**.

- **image(a, [x, y, w, h], rotation)** – команда аналогичная предыдущей, но позволяющая задавать поведение рисунка при вращении пиктограммы. Значение параметра **rotation** равное **'on'** позволяет поворачивать рисунок вместе с пиктограммой подсистемы.

- **patch(x, y)** – отображение закрашенного многоугольника, координаты которого заданы векторами **x** и **y**. Цвет рисунка – черный.

- **patch(x, y, [r g b])** - команда аналогичная предыдущей, но позволяющая задавать цвет рисунка. Параметры **r,g** и **b** задают соотношение красного, зеленого и синего цветов в рисунке (в черно-белом варианте этого не видно,

но цвет треугольника справа на Рис. 70 должен быть розовым). Значение параметров должно находиться в пределах от 0 до 1.

Примеры маскированных подсистем с различными вариантами команд отображения рисунков показаны на Рис. 70.

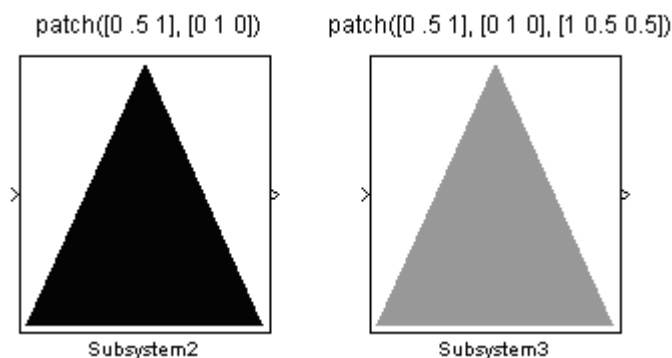


Рис. 70. Варианты отображения рисунков на пиктограммах

Параметры подсистемы создаются с помощью вкладки **Parameters** редактора маски (Рис. 71). Для создания поля ввода параметра с его описанием необходимо выполнить следующие действия:

1. Нажать кнопку **Add** (добавить).
2. Ввести описание параметра в поле **Prompt** (подсказка). В качестве описания параметра обычно используется его название в виде текста, например, “**Gain**”, “**Constant value**” и т.п.
3. Указать идентификатор параметра в поле **Variable** (переменная). Естественно, что это должен быть один из тех идентификаторов, который использовался при задании параметров блоков внутри подсистемы (хотя это не обязательно, поскольку параметр может быть использован и для модификации самого окна диалога). Все переменные, идентификаторы которых заданы на вкладке **Initialization**, помещаются в **Mask Workspace** – локальную рабочую область маски и являются доступными только внутри подсистемы.
4. Выбрать тип элемента интерфейса задающего параметр из списка **Type**:
 - **Edit** – редактируемое поле ввода.
 - **Checkbox** – флажок.
 - **Popup** – раскрывающийся список. В этом случае становится доступным для редактирования окно **Popups** (Элементы списка). В него необходимо ввести элементы списка, разделенные символом вертикальной черты (или каждый элемент на отдельной строке). Например, выражение **alpha|beta|gamma** задаст список из трех элементов: **alpha**, **beta** и **gamma**.
5. Выбрать формат параметра (флажок):

- Если флажок установлен: **Evaluate** – вычисляемый. Выбирается, если параметр должен иметь числовое значение. В данное поле можно будет ввести выражение в соответствии с правилами языка **MATLAB**. Формат **Evaluate** позволяет также использовать числовую форму значения переменной в том случае, если тип элемента интерфейса выбран в виде флажка или раскрывающегося списка. Так, например, для раскрывающегося списка **alpha|beta|gamma** значение связанной со списком переменной будет равно **1**, если в списке выбрано **alpha**, **2** – если в списке выбрано **beta**, и **3** – если в списке выбрано **gamma**. Для элемента интерфейса **Checkbox** вычисляемые значения будут равны **1** (при установленном флажке) и **0** (при снятом флажке).

- Если флажок не установлен – текстовый. Выбирается, если параметр должен быть строкой символов.

- Если установлен флажок **Tunable**, то этот параметр может быть изменен пользователем в процессе вычисления модели. Эта команда игнорируется для маскированного блока-источника, т. е. блока, имеющего только выходные порты.

6. Установить дополнительные опции для некоторых параметров:

Если флажок **Show parameter** установлен (по умолчанию), то выбранные параметры появляются в маскированном окне параметров.

Если убрать флажок **Enable parameter**, то пользователь не сможет менять значение параметра.

7. Если потребуется, ввести команды в окне **Callback**.

Команды инициализации представляют собой обычные команды на языке **MATLAB** и могут включать операторы и **m**-функции. Такие команды задают переменные, которые будут находиться в рабочей области маскированной подсистемы. Эти переменные доступны внутри подсистемы и могут быть использованы в качестве параметров блоков входящих в состав подсистемы, а также для создания пиктограммы подсистемы. Команды инициализации выполняются в следующих случаях:

- При открытии окна модели.
- При запуске модели на выполнение.
- При выполнении команды **Edit/Update diagram**.
- При вращении блока маскированной подсистемы (в этом случае команды инициализации обеспечивают перерисовку пиктограммы).
- При автоматическом изменении пиктограммы, зависящей от параметров блока.

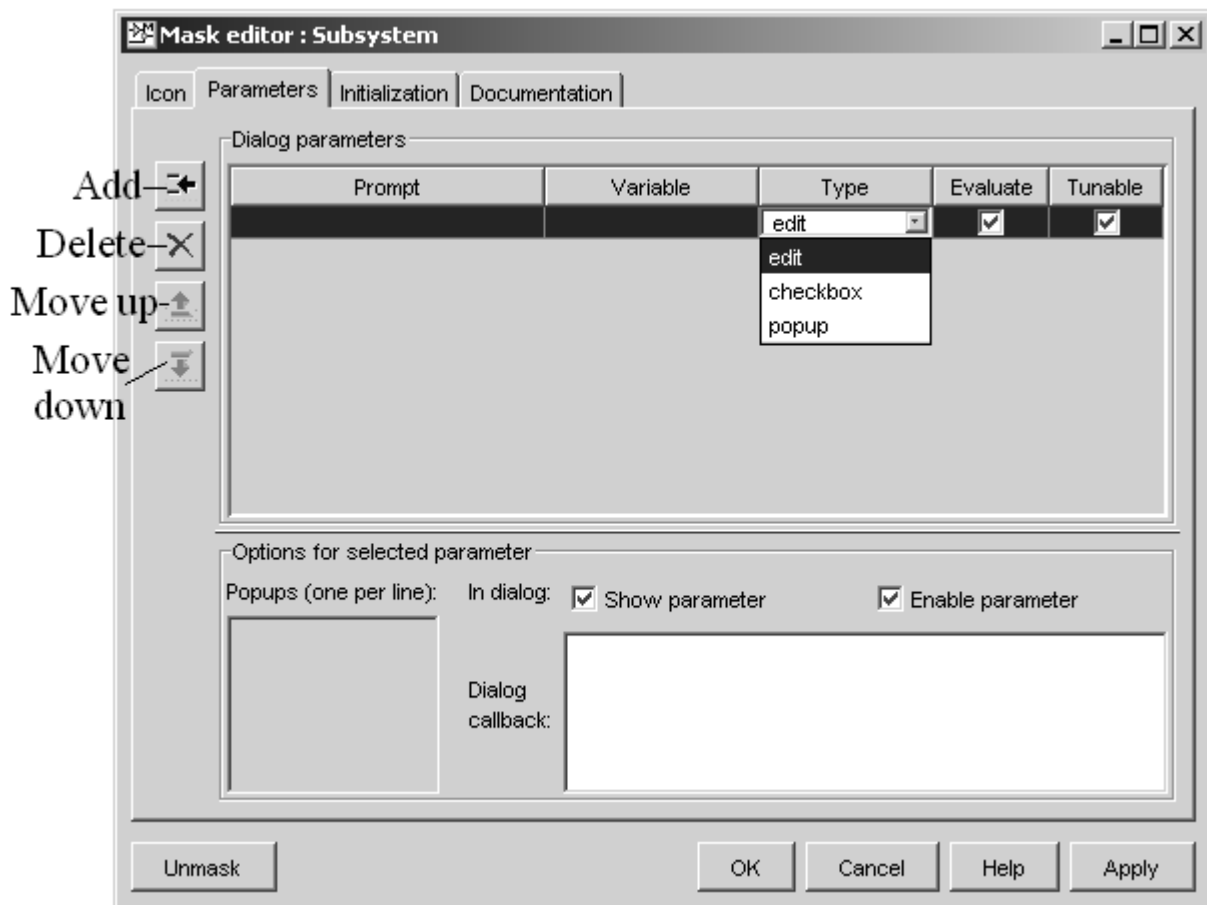


Рис. 71. Окно редактора маски: вкладка Parameters

Для создания описания и справки маскированной подсистемы служит вкладка **Documentation** (Документация). Вкладка **Documentation** содержит три графы: **Mask Type**, с помощью которого можно задать имя блока, **Mask description** (Описание блока) и **Mask Help** (Справка по блоку). Внешний вид редактора маски с открытой вкладкой **Documentation** показан на Рис. 73.

Текст, введенный в графу **Mask description**, отображается в верхней части окна диалога параметров блока и предназначен для краткого описания блока. В графу **Mask Help** вносятся команды, обеспечивающие загрузку файлов справки, созданных пользователем, в справочную систему при нажатии клавиши **Help** в окне параметров. Эти команды описаны в документации по **Simulink**. Наиболее удобным форматом файла справки является **htm** (**html**) – формат. Вызов справочного **htm**-файла осуществляется командой вида:

```
web(['file:/// ' which('helpfile.htm')]); ,
```

где

helpfile.htm – имя файла справки.

Для правильной работы справочной системы необходимо, чтобы файл справки находился в той же папке, что и файл модели, и данная папка являлась рабочей. Допускается также вместе с именем файла указывать его полный путь.

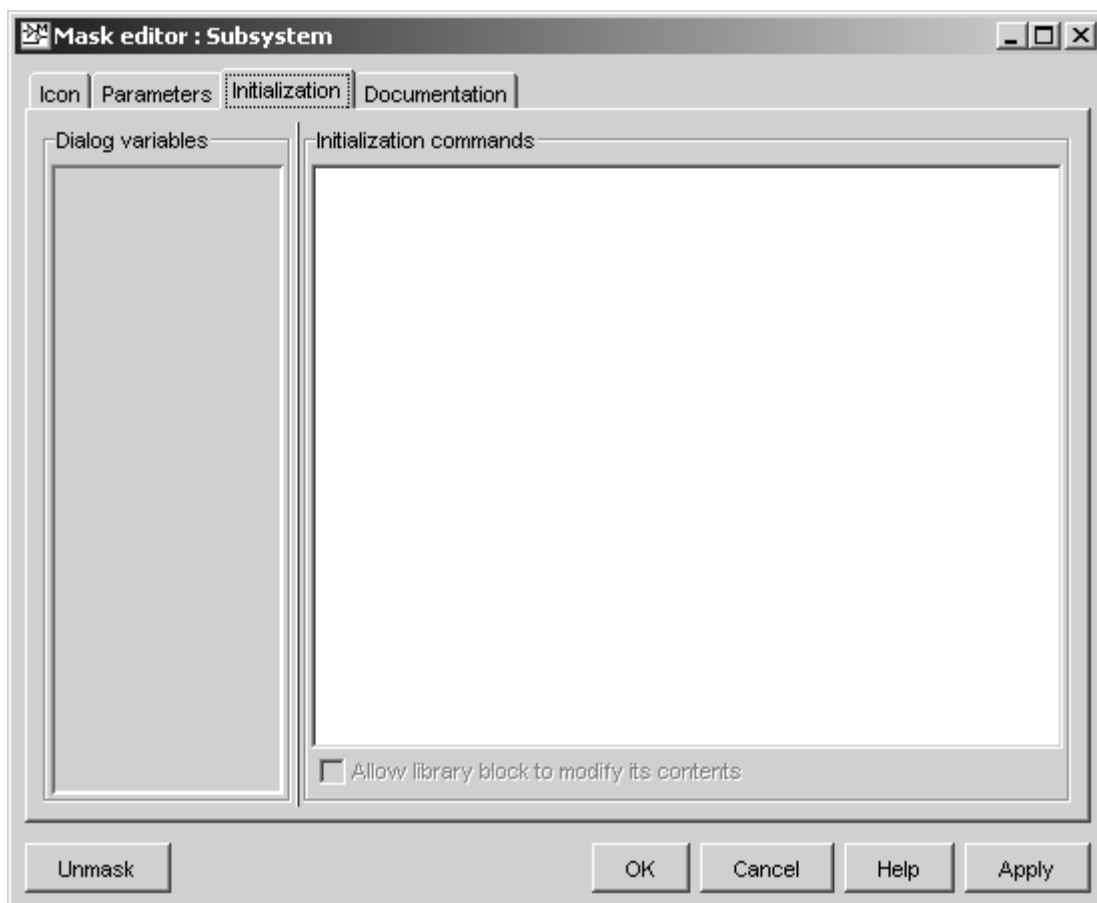


Рис. 72. Окно редактора маски: вкладка Initialization

Сделаем маску уже известной нам модели колебательного контура с подсистемой (модель **mask_subsystem.mdl** в приложении). Эту систему можно описывать через параметры элементов электрической цепи (R , L , C и начальные условия) или через коэффициенты дифференциального уравнения (ω_0 , β и начальные условия). Выберем второй способ. Параметрами, которые можно будет изменять в этом блоке, будут:

- собственная частота колебательного контура ω_0 – обозначается ω_0 ,
- коэффициент диссипации β – обозначается β ,
- ток в цепи в начальный момент времени $I(0)$ – обозначается I ,
- производная тока в начальный момент времени $I'(0)$ – обозначается dI .

Выполним команду **Edit/Mask subsystem** в окне модели. В появившемся окне **Mask Editor** на вкладке **Parameters** введем выбранные нами параметры. Эта вкладка изображена на Рис. 74.

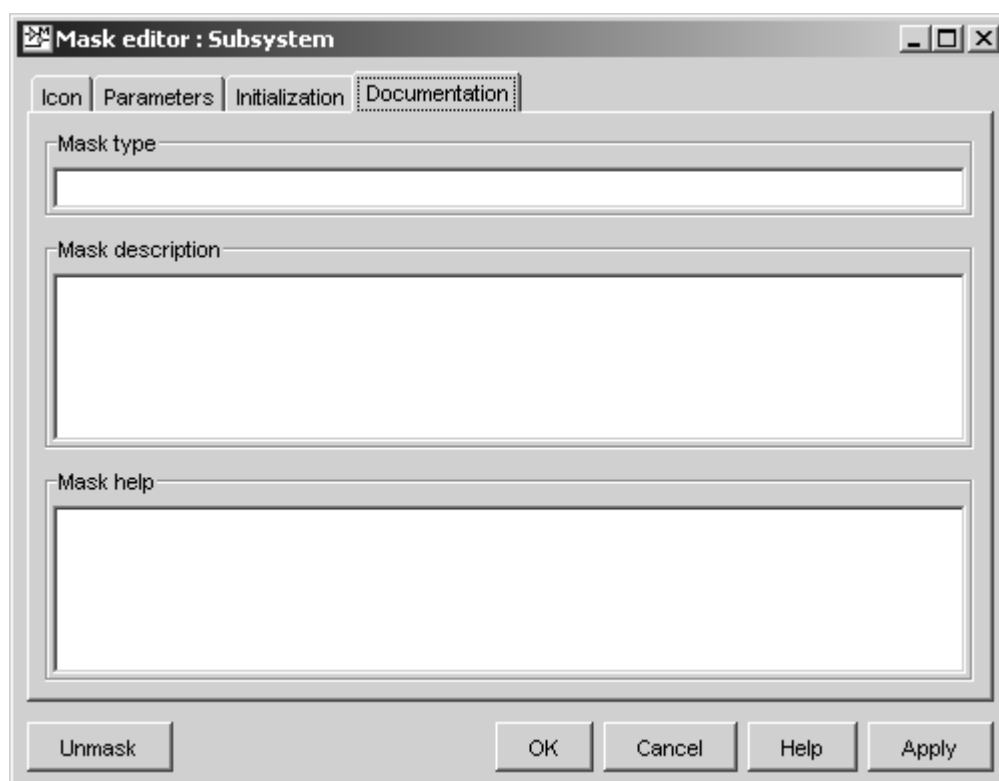


Рис. 73. Окно редактора маски: вкладка Documentation

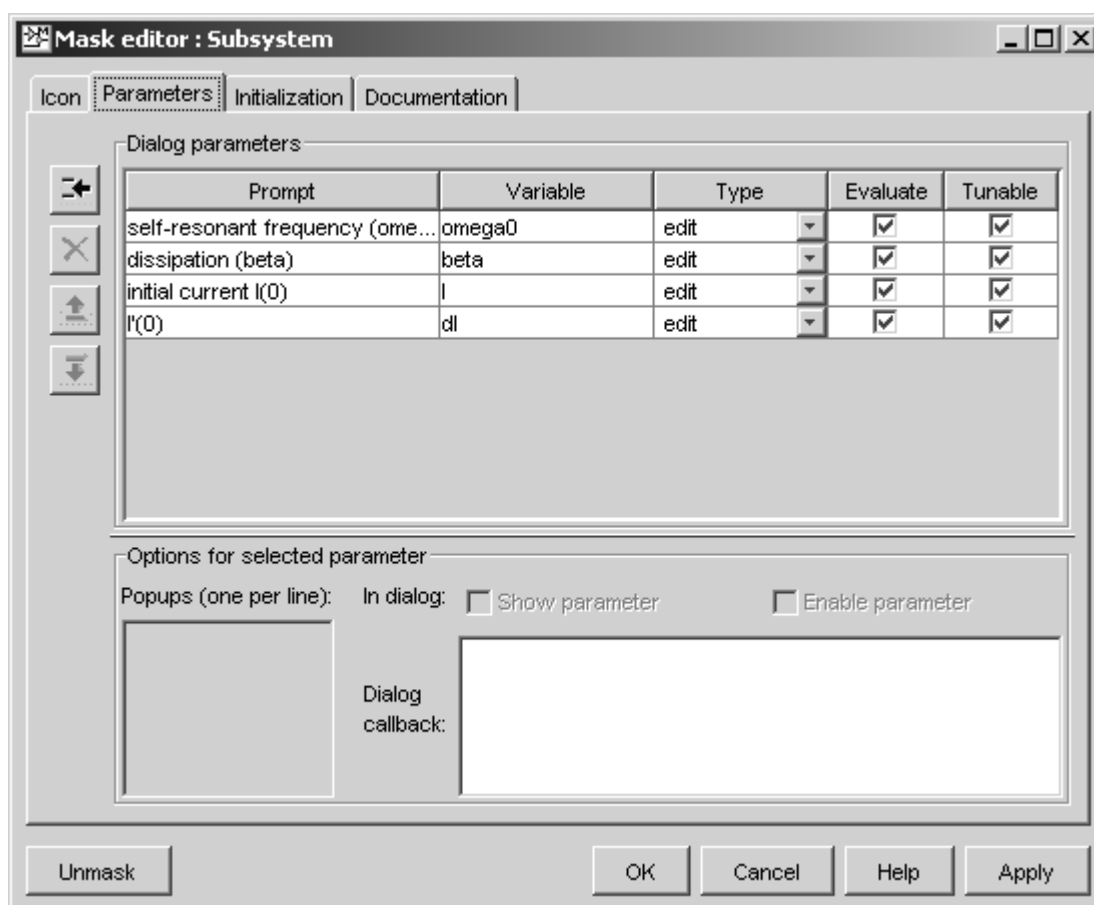


Рис. 74. Маскирование: вкладка параметров

В окнах параметров блоков подсистемы числовые выражения заменим на выбранные обозначения для параметров (или на их алгебраические выражения), как показано на Рис. 75.

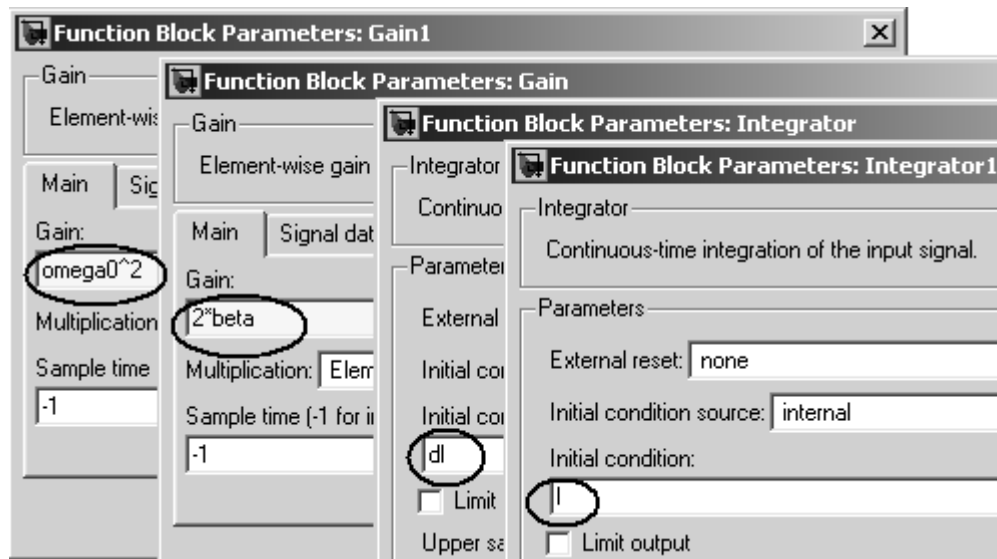


Рис. 75. Маскирование: обозначения для параметров блоков

На вкладке **Icon** в окне **Drawing commands** введем команду **disp('text')**, которая будет выводить текст внутри пиктограммы блока подсистемы (Рис. 76). Формат этой команды и образец пиктограммы изображены внизу окна. Также можно использовать команду **text(x, y, 'text')**, где x, y – координаты начала текста.

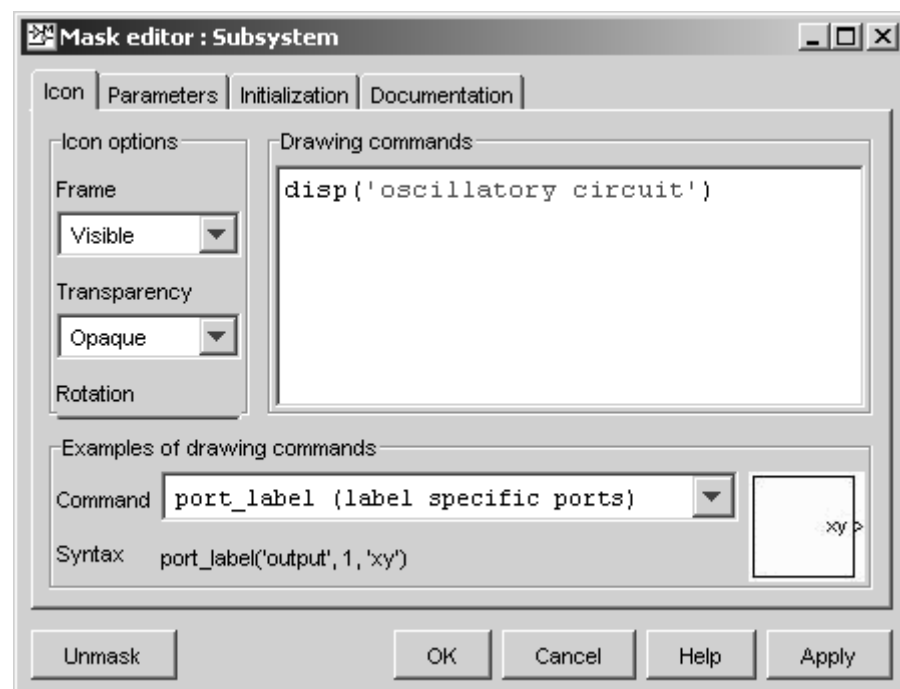


Рис. 76. Маскирование: вкладка пиктограмма

После применения команды **Apply** внешний вид пиктограммы блока подсистемы изменится (Рис. 77).

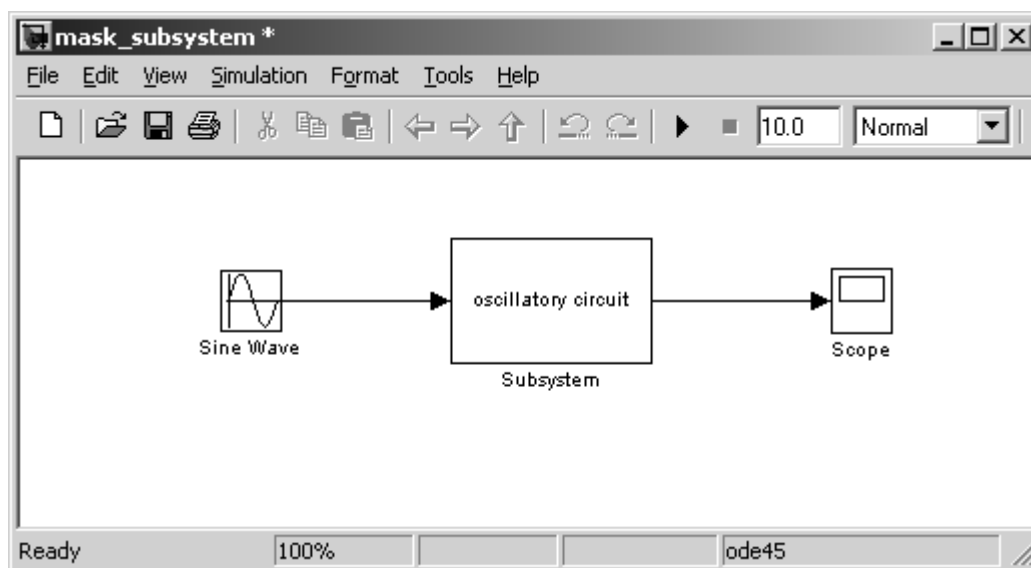


Рис. 77. Маскирование: модель

На вкладке документации введем пояснения для блока (Рис. 78).

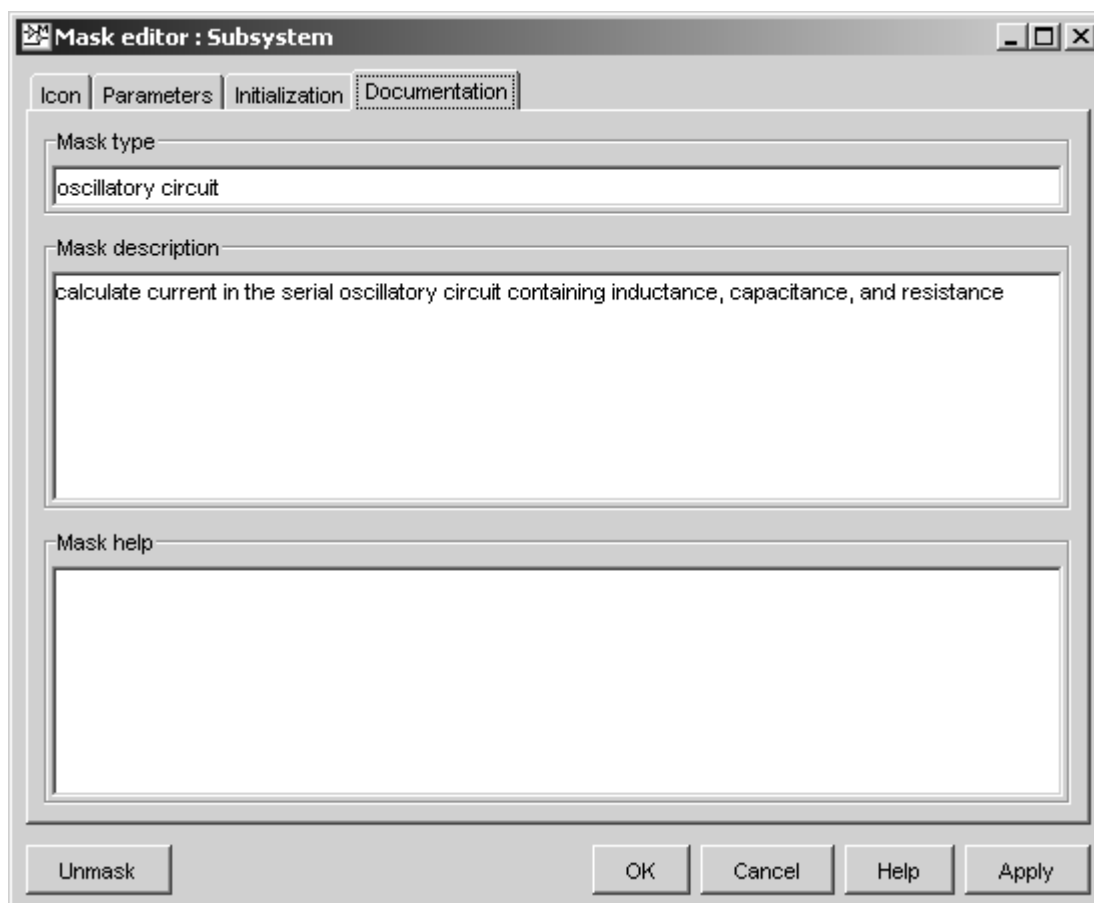


Рис. 78. Маскирование: вкладка документация

После подтверждения проделанных операций двойной щелчок по блоку подсистемы будет открывать окно параметров этого блока (Рис. 79), а не саму подсистему.

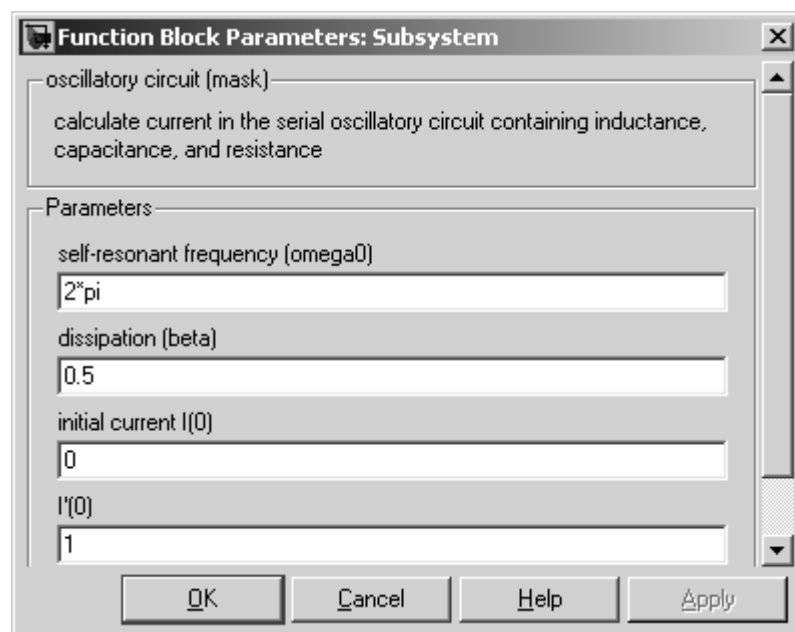


Рис. 79. Маскирование: окно параметров подсистемы

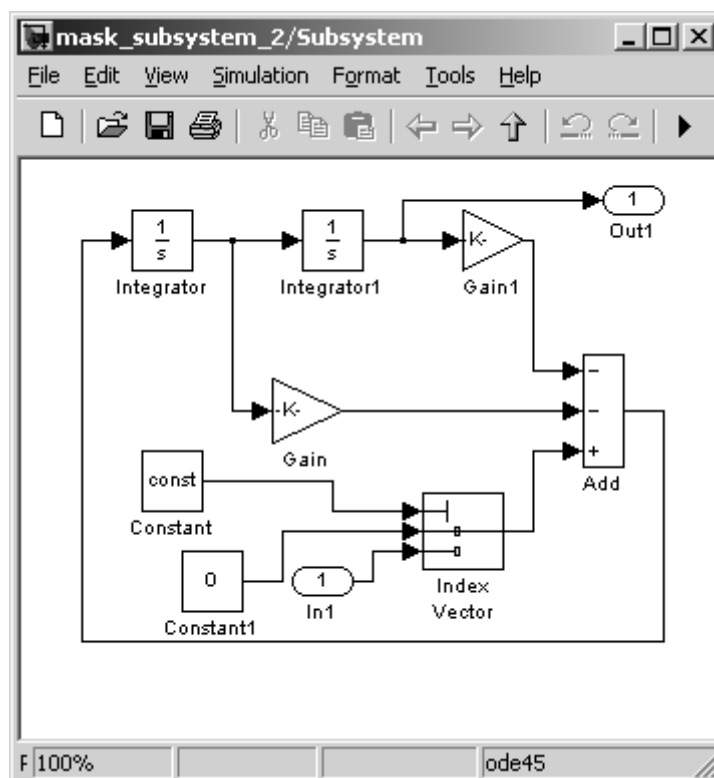


Рис. 80. Окно подсистемы модели колебательного контура. В модели можно выбрать, действует ли вынуждающая сила.

Чтобы открыть окно подсистемы нужно применить команды **Edit/Look under mask**.

Можно изменить эту модель, добавив возможность выбирать, действует ли на систему внешняя сила. Для этого можно добавить в схему подсистемы блоки, показанные на Рис. 80.

Здесь добавлен блок переключателя **Index Vector**. Окно параметров этого блока изображено на Рис. 81. Блок переключателя имеет несколько входов (в нашем случае 3). На верхний вход подается управляющий сигнал, значение которого осуществляет переключение между остальными входами. Если в окне параметров блока отмечен пункт **Use zero-based indexing**, следовательно, остальные входы будут нумероваться сверху вниз, начиная с нуля (в нашем случае вход «0» и вход «1»). Когда на управляющий вход подается 0, через переключатель проходит сигнал, подаваемый на вход «0». Когда на управляющий вход подается 1 – сигнал, подаваемый на вход «1».

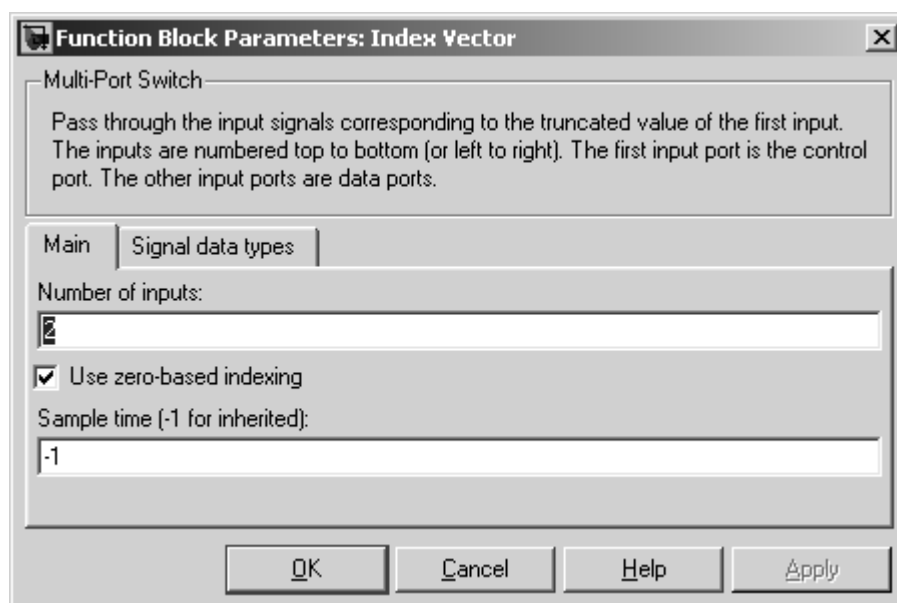


Рис. 81. Окно параметров блока переключателя

На вход «0» подается нулевой сигнал с блока «Источник постоянного сигнала», на вход «1» подается сигнал с входного порта (с генератора синусоидального сигнала). На управляющий вход подается сигнал с «Источника постоянного сигнала», но значение этого сигнала задано переменной **const** (Рис. 82).

Эту переменную **const** нужно добавить в окне параметров редактора маски (Рис. 83) в виде флажка **checkbox**.

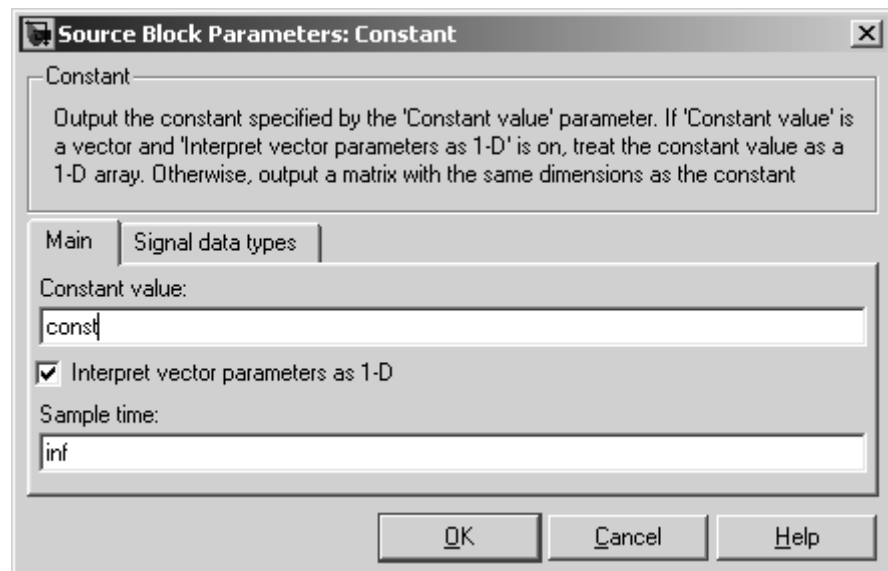


Рис. 82. Окно параметров «Источника постоянного сигнала»

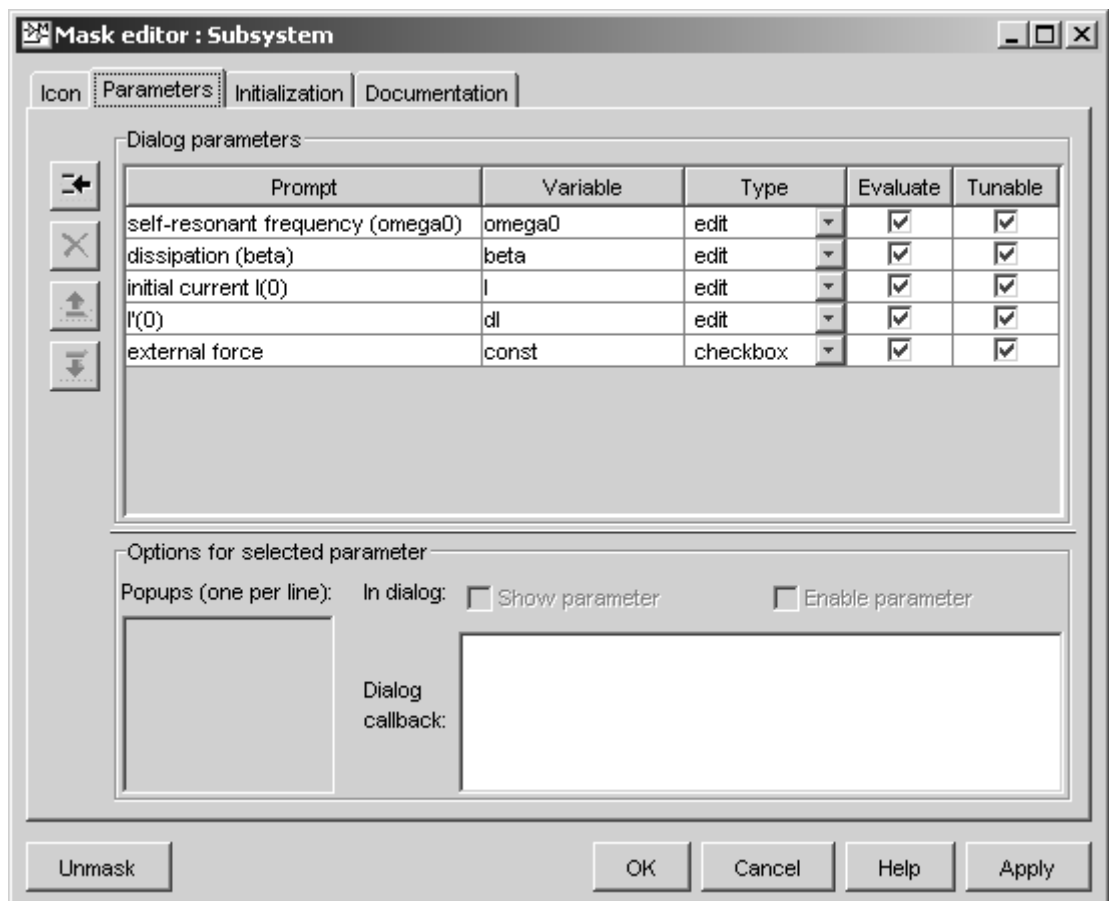


Рис. 83. Окно параметров редактора маски

Тогда в окне параметров подсистемы появится еще один пункт **external force** (Рис. 84).

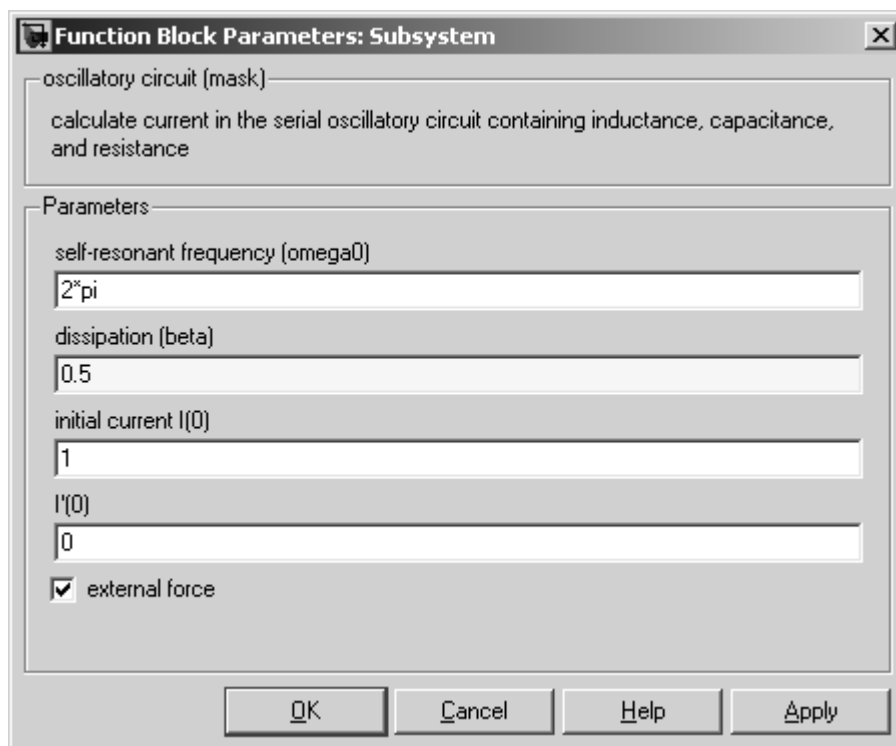


Рис. 84. Окно параметров подсистемы

Когда флажок **external force** установлен, переменная **const** принимает значение равное 1, управляющий сигнал переключателя равен 1, и через переключатель проходит сигнал со входа «1». Если же флажок **external force** не установлен, то **const** = 0, и через переключатель проходит сигнал со входа «0».

Использование редактора пиктограмм iconedit

Для создания пиктограмм можно также использовать редактор пиктограмм **iconedit**. Для его вызова используется команда:

iconedit('modelname','Subsystem'),

где **modelname** – имя файла модели (без расширения),

Subsystem – имя подсистемы, для которой будет создаваться пиктограмма.

Пиктограмма создается по точкам, расположение которых указывается с помощью мыши. Между собой точки соединяются прямыми линиями. Для того, чтобы начать новую линию, необходимо нажать клавишу **n** на клавиатуре. Для отмены создания последней точки используется клавиша **d**. Выход из редактора с автоматическим обновлением пиктограммы осуществляется клавишей **q**. По завершении работы с редактором необходимо также закрыть его окно рисования. Кроме обновления пиктограммы завершение работы с

редактором пиктограмм сопровождается выводом в командной строке **MATLAB** графической команды, обеспечивающей построение пиктограммы.

Пример пиктограммы, созданный с помощью **iconedit**, его окно рисования показаны на Рис. 85.

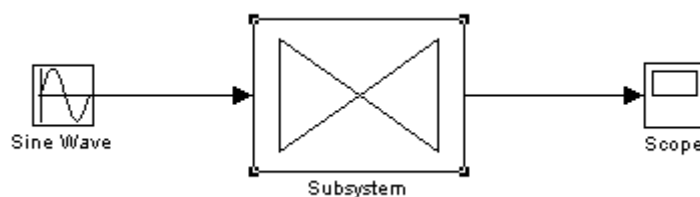
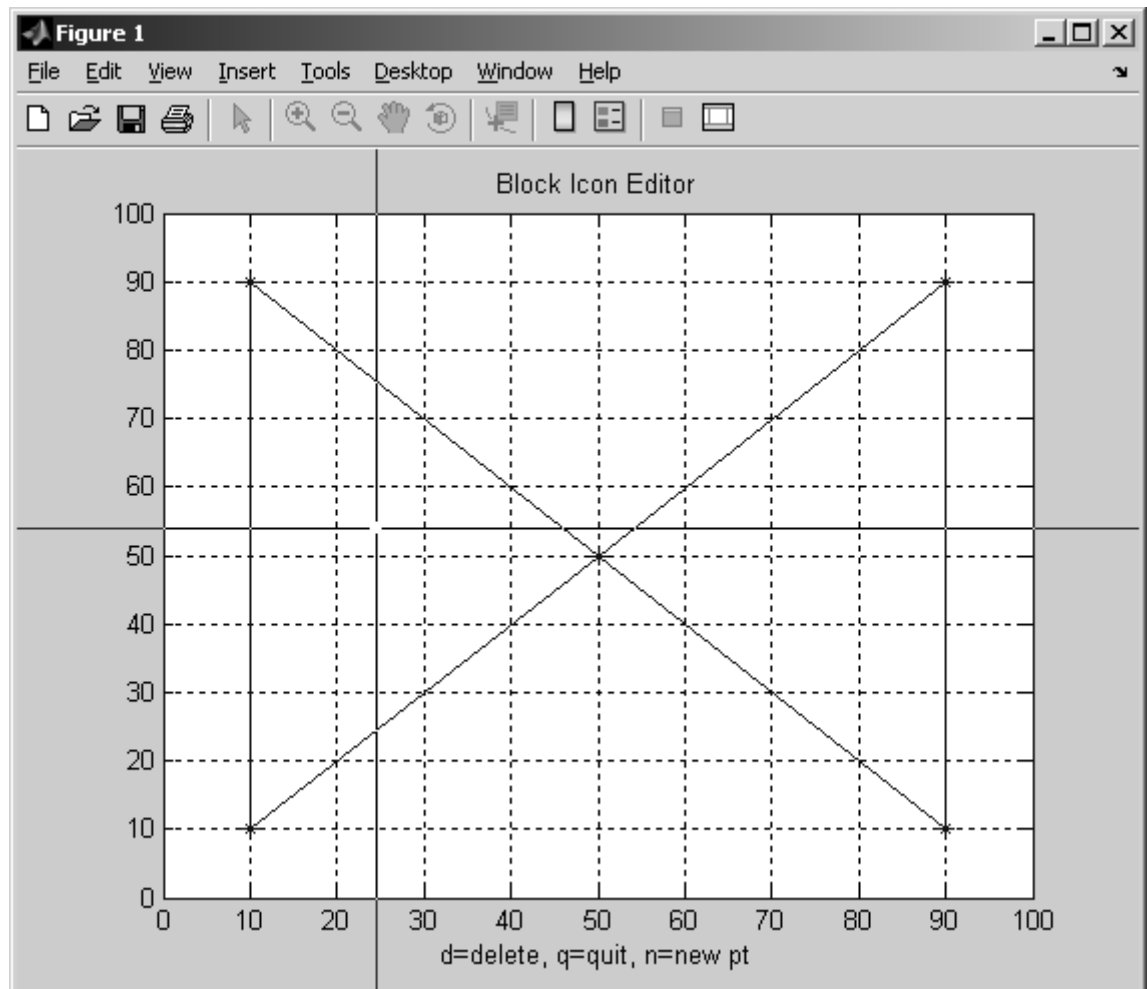


Рис. 85. Создание пиктограммы с помощью iconedit

В рабочем окне **MATLAB** появляется текст команды, обеспечивающей построение пиктограммы:

ans =

```
plot(0,0,100,100,[10,50,10,10],[90,50,10,90],[90,90,50,90],[90,10,50,90])
```

2.7. Использование Simulink LTI-Viewer для анализа динамических систем.

Инструмент **Simulink LTI-Viewer** входит в состав пакета прикладных программ **Control System Toolbox** и предназначен для анализа линейных стационарных систем. С помощью данного инструмента можно легко построить частотные характеристики исследуемой системы, получить ее отклики на единичные ступенчатое и импульсное воздействия, найти нули и полюса системы и т.д. Рассмотрим работу с **Simulink LTI-Viewer**.

Если система задана с помощью одного блока как, например, для рассмотренного нами ранее фильтра нижних частот, определенного с помощью блока передаточной функции, то достаточно, выделив блок, из контекстного меню выбрать пункт **Linearize Block...** В появившемся окне **Control and Estimation Tools Manager** нажать кнопку **Linearize Block**. Данная команда выполняет линеаризацию модели (загружает инструмент **LTI-Viewer**) и строит реакцию системы на единичное ступенчатое воздействие. Как изменить настройки окна **LTI-Viewer** и получить другие характеристики системы, покажем на другом примере.

Рассмотрим случай, когда **Simulink**-модель состоит из нескольких блоков. Например, последовательно включенные блоки передаточных функций интегрирующей и дифференцирующей цепочек. Заметим, что такое соединение блоков не даёт тот же результат что последовательное соединение двух цепочек на практике для электрических цепей. Последовательное соединение двух блоков передаточной функции означает, что сначала будет решено дифференциальное уравнение, соответствующее первой передаточной функции, а потом полученное решение будет использовано как входной сигнал при решении второго уравнения. В электрических цепях ток, вызываемый входным напряжением, будет делиться между первой и второй цепочкой, поэтому коэффициент передачи не равен произведению коэффициентов передачи отдельных цепочек.

Для блоков передаточной функции выберем следующие параметры:

	числитель	Знаменатель
Интегрирующая цепочка	[1]	[1e-1 1]
Дифференцирующая цепочка	[1e-1 0]	[1e-1 1]

После построения модели для начала работы с **LTI-Viewer** нужно указать вход и выход исследуемой части системы. Для этого нужно выделить линию связи на входе в систему, правой кнопкой мыши открыть контекстное меню и выбрать команды **Linearization Points/Input Point**. Над линией связи появится метка, обозначающая вход в систему (**Input Point**). Аналогично, нужно выделить линию связи на выходе из системы и выбрать команды **Li-**

nearization Points/Output Point. На Рис. 86 изображена **Simulink**-модель с обозначенными точками входа и выхода в систему.

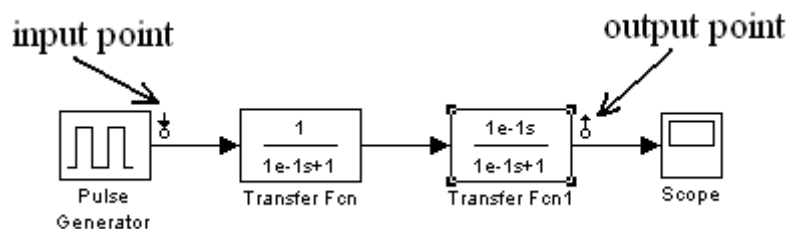


Рис. 86. Точки входа и выхода для линеаризации системы

Далее следует выбрать в меню **Tools/Control Design/Linear Analysis**, и в появившемся окне **Control and Estimation Tools Manager** (Рис. 87) нажать кнопку **Linearize Model**.

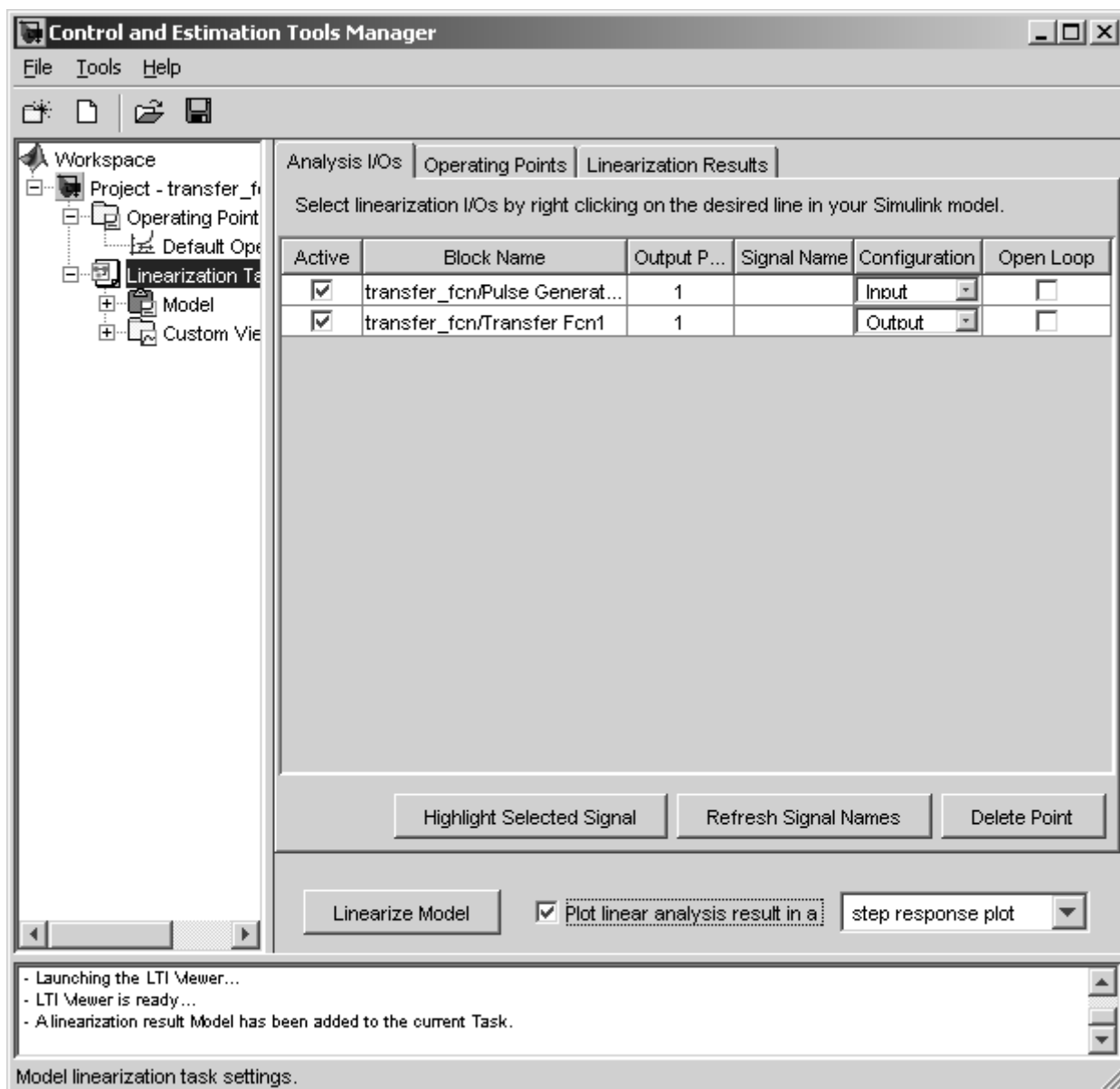


Рис. 87. Окно Control and Estimation Tools Manager

Появится окно **LTI-Viewer** с графиком реакции системы на единичное ступенчатое воздействие (переходной характеристикой). Результат выполнения данного пункта показан на Рис. 88.

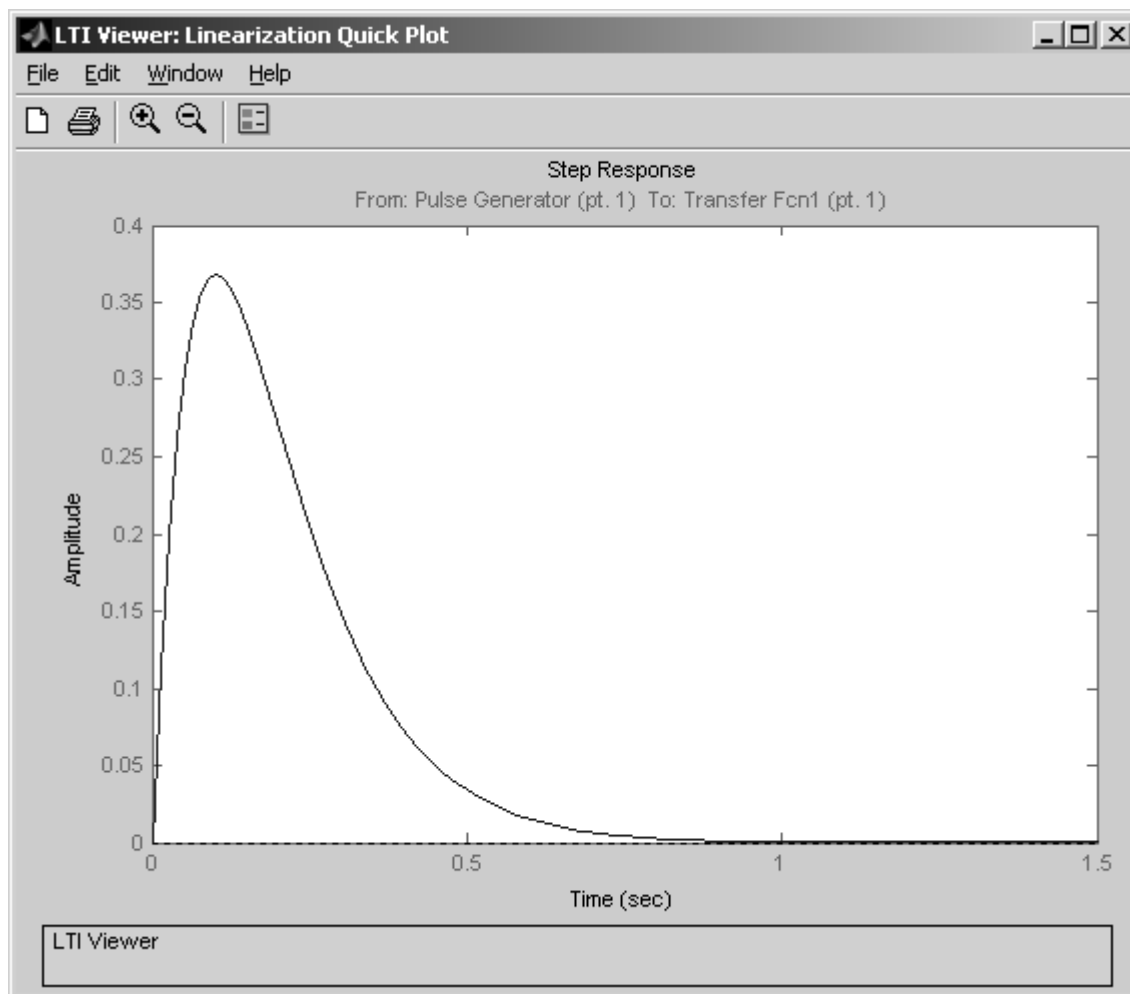


Рис. 88. Реакция системы на единичное ступенчатое воздействие

Для получения остальных характеристик системы необходимо выполнить команду **Edit\Plot Configuration...** в окне **LTI Viewer**. В результате выполнения этой команды откроется окно **Plot Configurations** (Рис. 89).

В открывшемся окне можно выбрать число отображаемых графиков (панель **Select a response plot configuration**) и вид отображаемых графиков (панель **Response type**).

Для построения доступны следующие графики (диаграммы):

- **step** – реакция на единичное ступенчатое воздействие,
- **impulse** – реакция на единичное импульсное воздействие,
- **bode** – логарифмические амплитудная и фазовая частотные характеристики,

- **bode mag** – логарифмическая амплитудная частотная характеристика,
- **nyquist** – диаграмма Найквиста,
- **nichols** – годограф Николса,
- **sigma** – сингулярные числа,
- **pole/zero** – нули и полюса системы.

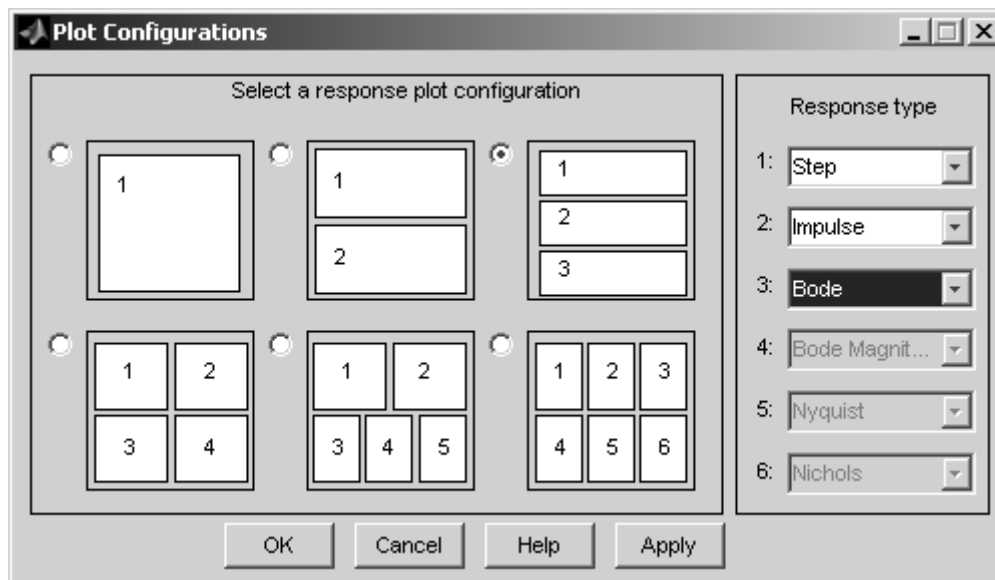


Рис. 89. Окно Plot Configurations

Ниже приведен пример окна **Simulink LTI-Viewer** с несколькими различными характеристиками исследуемой системы: импульсной характеристикой, АЧХ и ФЧХ (Рис. 90).

Настройку внешнего вида графиков можно выполнить с помощью команды **Edit\Line Styles...** (установка вида и цвета линий, вида маркеров). Поменять единицы измерения и настройки для графиков можно с помощью команды **Edit\Viewer Preferences....** Окно **LTI Viewer Preferences** имеет четыре вкладки (Рис. 91):

1. Вкладка **Units** окна позволяет задать единицы измерения частоты (**рад/с** или **Гц**), уровня (**dB** (децибелы) или абсолютные единицы), фазы (градусы или радианы), а также установить вид шкалы (логарифмический или линейный).
2. Установка стиля графиков (вкладка **Style**). На данной вкладке можно выполнить настройку шрифтов окна **Simulink LTI-Viewer** (панель **Fonts**), выбрать цвет осей графиков (панель **Colors**), а также задать нанесение линий сетки на графики (флажок **Show grids**).
3. Установка параметров расчета переходного процесса (вкладка **Characteristics**). Данная вкладка позволяет изменить параметры, установленные «по умолчанию» для вычисления времени нарастания и времени переходного процесса. По умолчанию **Simulink**

LTI-Viewer вычисляет время переходного процесса как время, когда переходная функция входит в 2% зону и больше не выходит из нее (параметр **Show setting time within**). Также можно изменить параметры для вычисления времени переходного процесса (**Show rise time from**). На данной вкладке имеется также флажок **Unwrap phase**, установка которого позволяет избежать отображения разрывов в фазо-частотной характеристике, связанных с областью определения функции **arctg**, вычисляющей фазовый сдвиг.

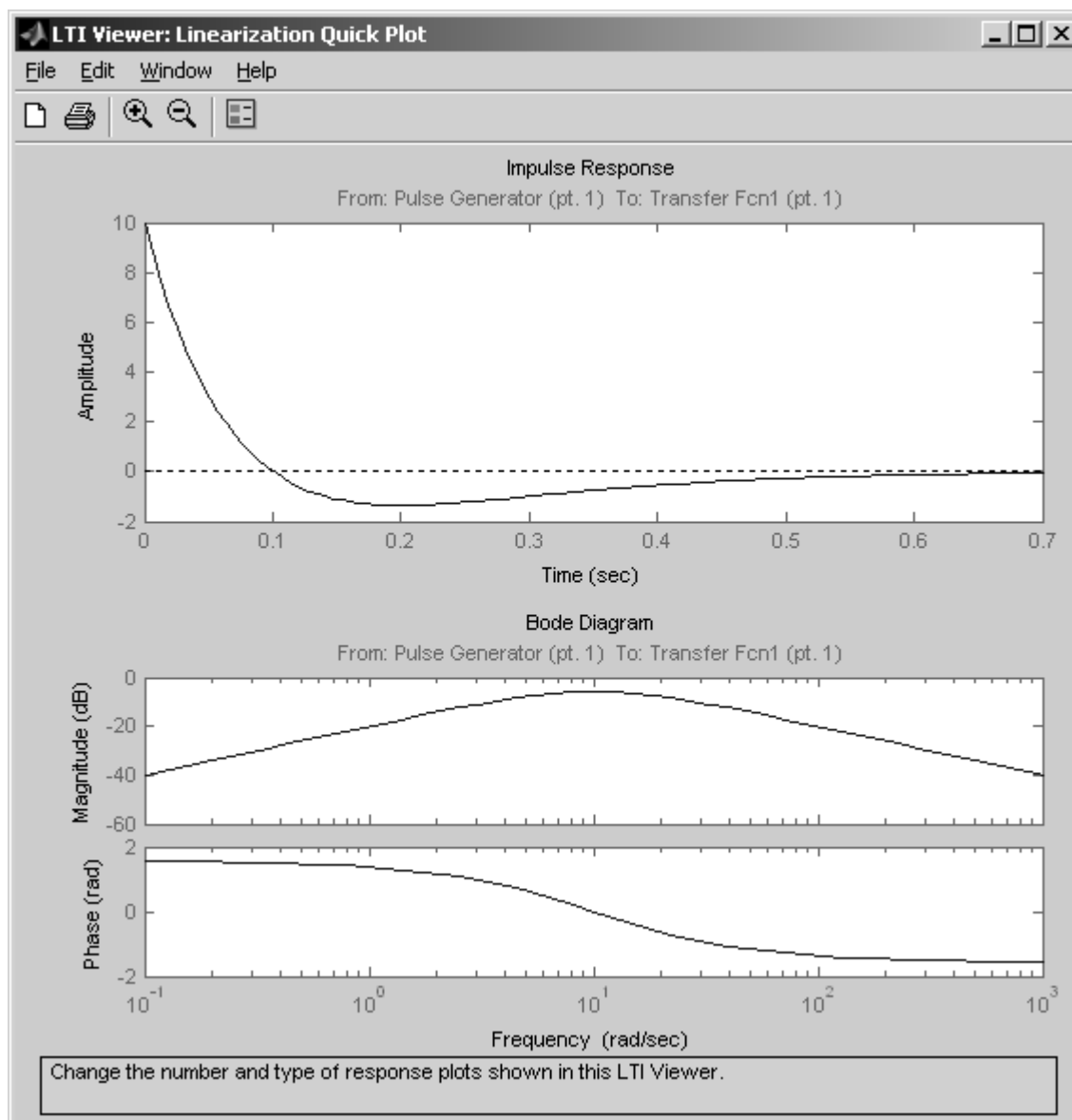


Рис. 90. Окно Simulink LTI-Viewer

4. Установка интервалов времени и частоты (вкладка **Parameters**). На данной вкладке задается временной интервал для расчета переходных процессов (панель **Time Vector**), а также интервал частот для расчета частотных характеристик (панель **Frequency**

Vector). Векторы времени и частоты можно вычислять в автоматическом режиме (**Generate automatically**), ввести конкретное значение для времени окончания расчета (**Define stop time**) или диапазон значений по частоте (**Define range**), либо задать непосредственно вектор значений времени или частоты (**Define vector**).

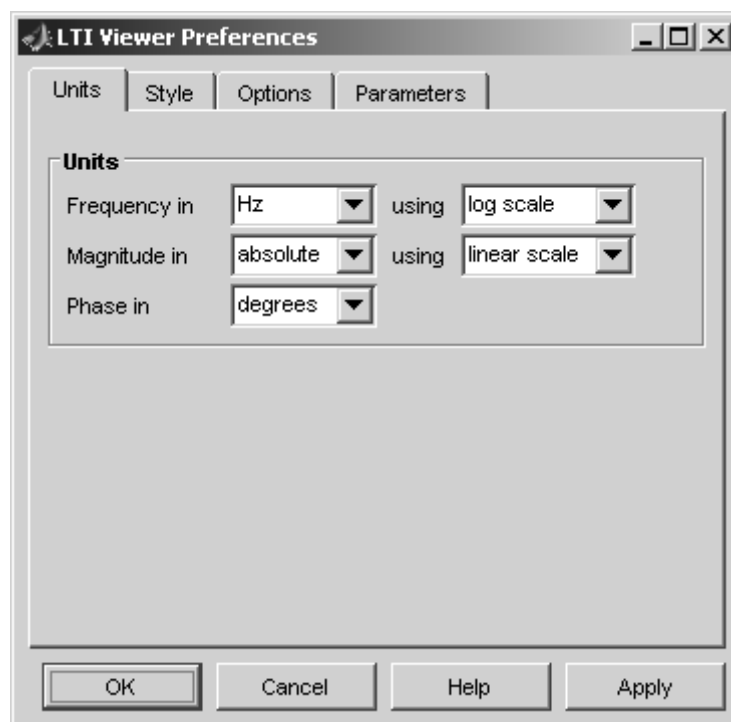


Рис. 91. Окно LTI Viewer Preferences

После изменения единиц измерения АЧХ и ФЧХ исследуемой системы будут выглядеть следующим образом (Рис. 92).

2.8. Начало работы в SimPowerSystems

SimPowerSystems содержит набор блоков для имитационного моделирования электротехнических устройств. В состав библиотеки входят модели пассивных и активных электротехнических элементов, источников энергии, электродвигателей, трансформаторов, линий электропередачи и другого оборудования. Имеется также раздел, содержащий блоки для моделирования устройств силовой электроники, включая системы управления для них.

Для начала работы нужно открыть **MATLAB**, как описывалось выше, открыть окно обозревателя разделов **Simulink** и выбрать среди списка разделов **SimPowerSystems**. Для загрузки этой библиотеки нужно нажать на пиктограмму свернутого узла (символ "+"), в правом окне откроется список основных разделов **SimPowerSystems** (Рис. 93). Некоторые из этих разделов, в

свою очередь, состоят из подразделов, но также в списке разделов могут находиться отдельные блоки, например, блок **Powergui**.

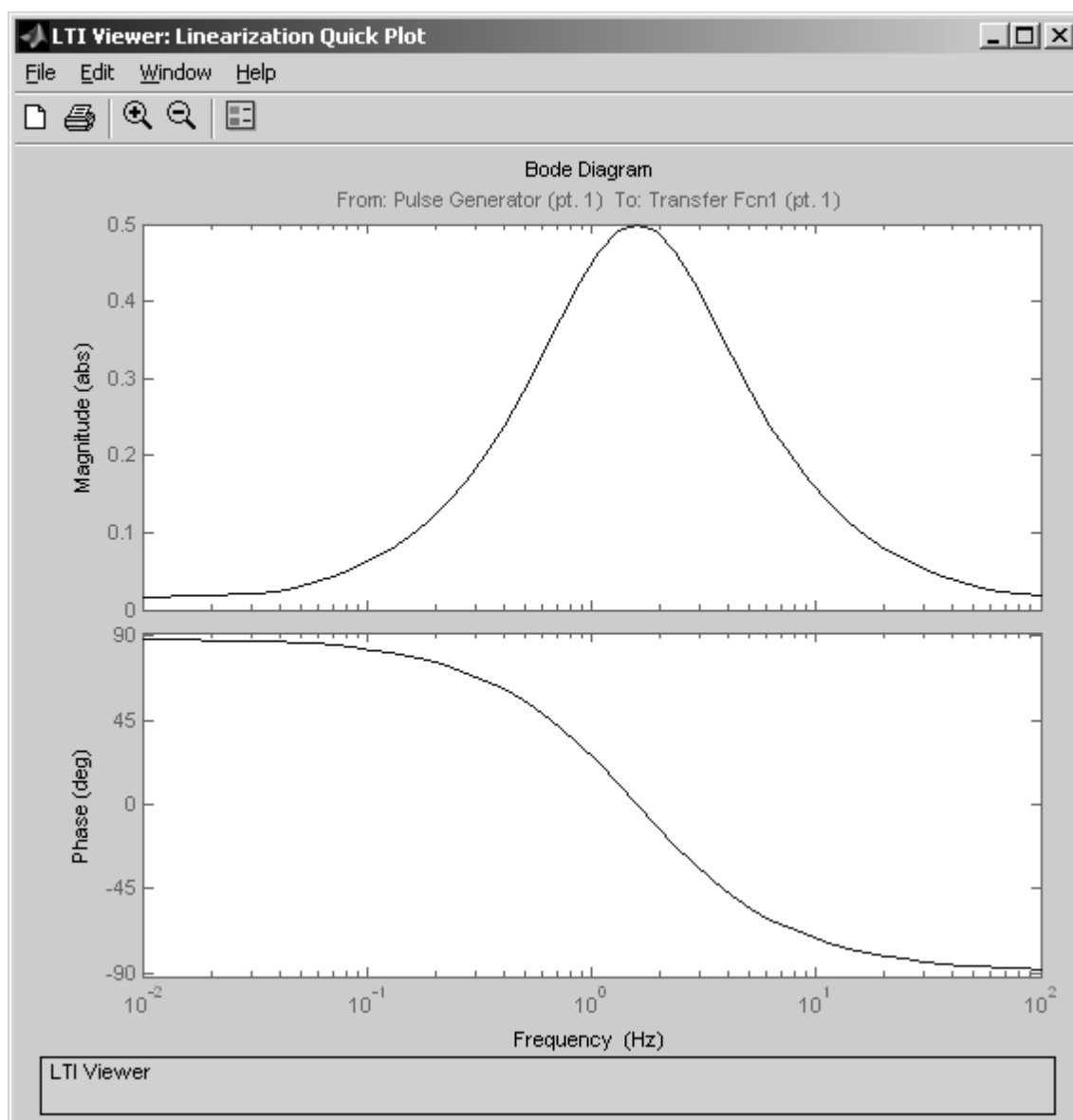


Рис. 92. АЧХ и ФЧХ исследуемой системы

Перечислим основные разделы библиотеки **SimPowerSystems**:

- **Extra library** – дополнительные электротехнические устройства;
- **Application libraries** – библиотека приложений;
- **Electrical Sources** – источники электрической энергии;
- **Elements** – электротехнические элементы;
- **Machines** – электрические машины;
- **Measurements** – измерительные и контрольные устройства;
- **Power Electronics** – устройства силовой электроники;
- **Powergui** – графический интерфейс пользователя пакета моделирования энергетических систем.

Методика создания модели **SimPowerSystems** (SPS-модели) ничем не отличается от методики создания модели на основе базовой библиотеки **Simulink**. Так же как и для обычной **Simulink**-модели, необходимо выполнить расстановку блоков на схеме, задать их параметры, соединить блоки и установить параметры расчета модели в целом.

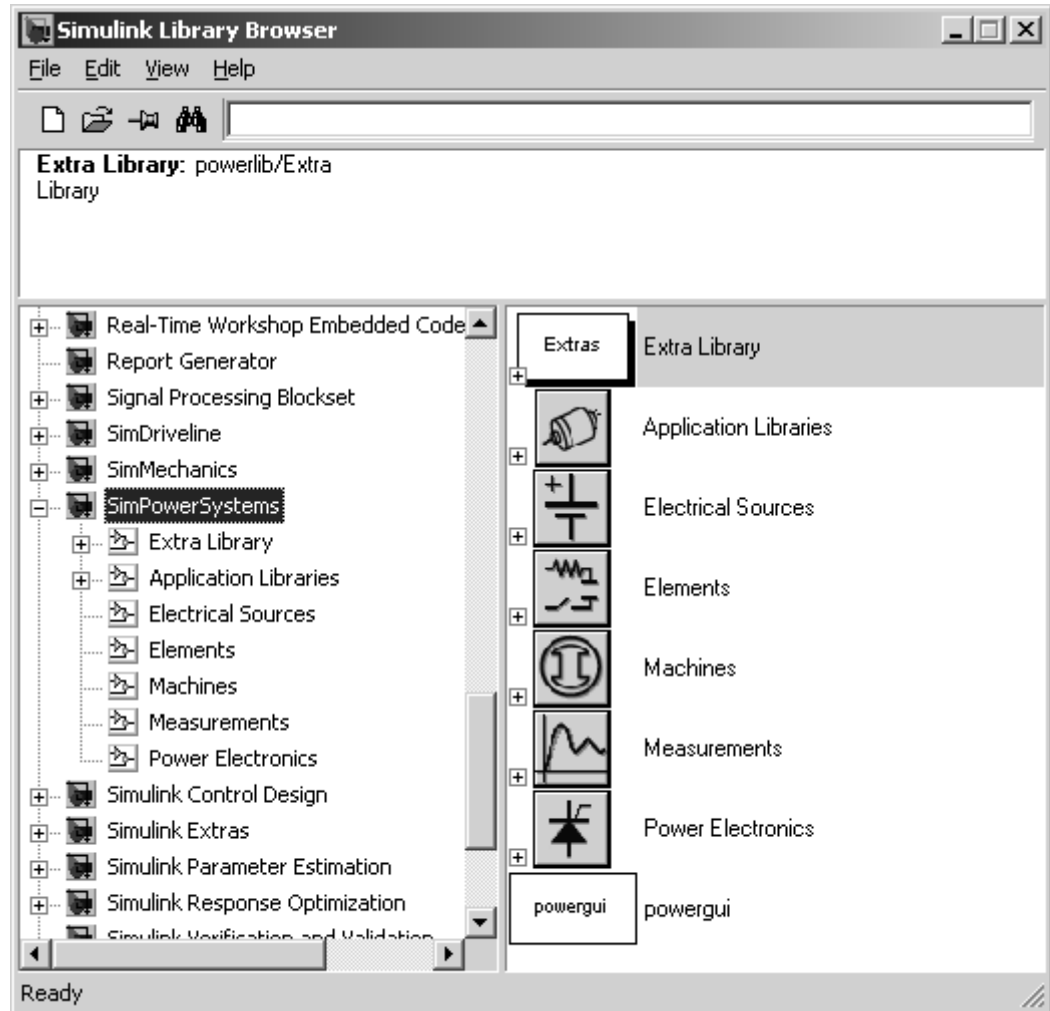


Рис. 93. Основные разделы библиотеки SimPowerSystems

Однако SPS-модели имеют и некоторые особенности:

1. Входы и выходы **SPS**-блоков, в отличие от блоков **Simulink**, не показывают направление передачи сигнала и имитируют соединительные клеммы электрических приборов. Таким образом, электрический ток может через вход или выход блока протекать в двух направлениях: как вовнутрь блока, так и наружу.
2. Соединительные линии между блоками являются, по сути, электрическими проводами, по которым ток может протекать также в двух направлениях. В **Simulink**-моделях же информационный сигнал распространяется только в одном направлении – от выхода одного блока к входу другого.

3. **Simulink**-блоки и **SimPowerSystems**-блоки не могут быть непосредственно соединены друг с другом. Сигнал от **S**-блока можно передать к **SPS**-блоку через управляемые источники тока или напряжения, а наоборот – с помощью измерителей тока или напряжения.
4. При расчете схемы, содержащей нелинейные блоки, следует использовать методы:
 - **ode15s** - многошаговый метод переменного порядка (от 1 до 5), использующий формулы численного дифференцирования,
 - **ode23tb** - неявный метод Рунге-Кутты в начале решения и метод, использующий формулы обратного дифференцирования 2-го порядка в последующем,которые дают наилучшие результаты по быстродействию.
5. Для расчета **SimPowerSystems**-модели необходим блок **Powergui**. Он может находиться в любой части окна модели и используется для хранения данных об эквивалентной **Simulink**-модели, построенной на основе модели пространства состояний электрической схемы. Если этот блок не был добавлен пользователем, то он добавляется автоматически при запуске модели на расчет.

Рассмотрим различие **Simulink**- и **SimPowerSystems**-моделей. Блоки из разных библиотек **Simulink** могут быть разделены на две категории по принципу своего действия. К одной категории относятся блоки, в которых непосредственно производится обработка информации. К другой категории относятся блоки, которые используются только для описания физической модели, а вычисления в процессе моделирования выполняются с помощью других средств. Например, к первой категории относятся блоки из основной библиотеки **Simulink**, **Signal Processing Blockset**, а ко второй категории относятся блоки из библиотек, предназначенных для моделирования различных физических систем (моделей) – это библиотеки **RF Blockset**, **SimDriveline**, **SimEvents**, **SimMechanics** и **SimPowerSystems**. Характерным отличием блоков из второй категории является наличие у них портов специального типа **Physical Modeling Connection Port** или другое название для этих портов **PMIOPort**. Порты, по которым в модели передаются числовые данные от блока к блоку, обозначаются стрелками, а порты, которые используются для объединения блоков в физические системы, например электрические схемы, обозначаются различным образом, в зависимости от того к какой библиотеке они относятся. В **SimPowerSystems** такие порты обозначаются квадратиками и, как уже говорилось, имитируют клеммы для подключения электрических устройств. Соответственно, нельзя соединять друг с другом порты различного типа. С помощью блоков управляемых источников тока и напряжения **Controlled Voltage Source** и **Controlled Current Source** сигналы, заданные числовыми данными, превращаются в соответствующие им значения тока и напряжения в электрической схеме. Для измерения тока и напряжения и получения их значений в численном виде (или для дальнейших операций над этими сигналами) используются блоки **Voltage Measurement** и **Current Measurement**.

Кратко рассмотрим алгоритм расчета **SimPowerSystems**-модели (Рис. 95). После того как **SPS**-модель построена (из блоков библиотеки **Simulink** и **SimPowerSystems**), она может быть запущена на расчет так же как любая модель **Simulink**. Перед началом каждого расчета происходит инициализация модели. В процессе инициализации происходит анализ электрической цепи, выявление ее топологии (наличие источников, приемников, нелинейных элементов и т.д.).

Для того чтобы **Simulink** смог выполнить моделирование, электрическая схема должна быть задана с помощью дифференциальных уравнений. Поэтому, в процессе анализа электрической цепи происходит проверка правильности соединения элементов, если есть короткие замыкания или разорванные участки, то выводится предупреждение. После этого нелинейные элементы заменяются на их эквивалентные модели, которые приводятся в справке к этим элементам. Эквивалентная модель нелинейного блока состоит из ключей, источников тока и напряжения, линейных элементов и блока, выполняющего логические функции и управляющего ключом и источниками. Для линейной части электрической схемы составляется система линейных дифференциальных уравнений в терминах пространства состояний, то есть в том виде, в котором их решает **Simulink**. Нелинейные блоки выступают как дополнительные источники управляемые сигналом (Рис. 94).

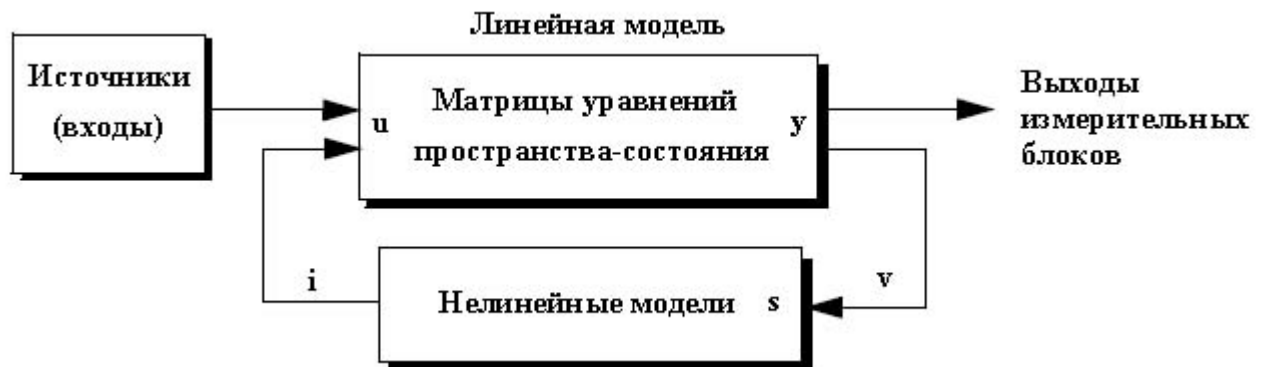


Рис. 94. Связь нелинейных моделей с **Simulink**-моделью

При выполнении моделирования в командном окне **MATLAB** могут выводиться предупреждения, в которых рекомендуется изменить настройки решателя. Обычно, при использовании нелинейных элементов и ключей в электрической схеме, рекомендуется использовать решатель с переменным шагом по времени (**continuous-time solver**). Для достижения большей точности в настройках решателя параметр **Type** задать **Variable-step**, а параметр **Solver** – **ode23s** (**ode23t**, **ode23tb**). Особенно внимательным надо быть к выбору ограничений шага по времени, так как **Simulink** не всегда правильно выбирает его автоматически. Желательно задавать параметр **Max step size** исходя из теоремы Котельникова.

Для моделирования электрических цепей в модели обязательно должен присутствовать блок **Powergui**. С помощью этого блока могут быть заданы

начальные условия для электрической цепи, построены временные и частотные характеристики линейных участков цепи. Для больших сложных электрических схем рекомендуется производить дискретизацию модели (с помощью блока **Powergui**), тогда решение ищется с постоянным шагом по времени, за счёт этого значительно может быть сокращено время моделирования.

В диалоговом окне блока **Powergui** есть возможность выбора способа моделирования с переменным шагом по времени (**Continuous-time**) либо с постоянным шагом (**Discretize electrical model**). В случае постоянного шага по времени в том же окне задаётся параметр **Sample time**. Выбор постоянного или переменного шага по времени в блоке **Powergui** действуют независимо от настроек заданных в диалоговом окне **Configuration parameters** модели, так как анализ и моделирование электрической цепи выполняется не самим **Simulink** непосредственно, а другими средствами в библиотеке **SimPowerSystems**, предназначенными для моделирования физических систем. **SimPowerSystems** в данном случае выступает в роли графического интерфейса, для того, чтобы собрать электрическую цепь.

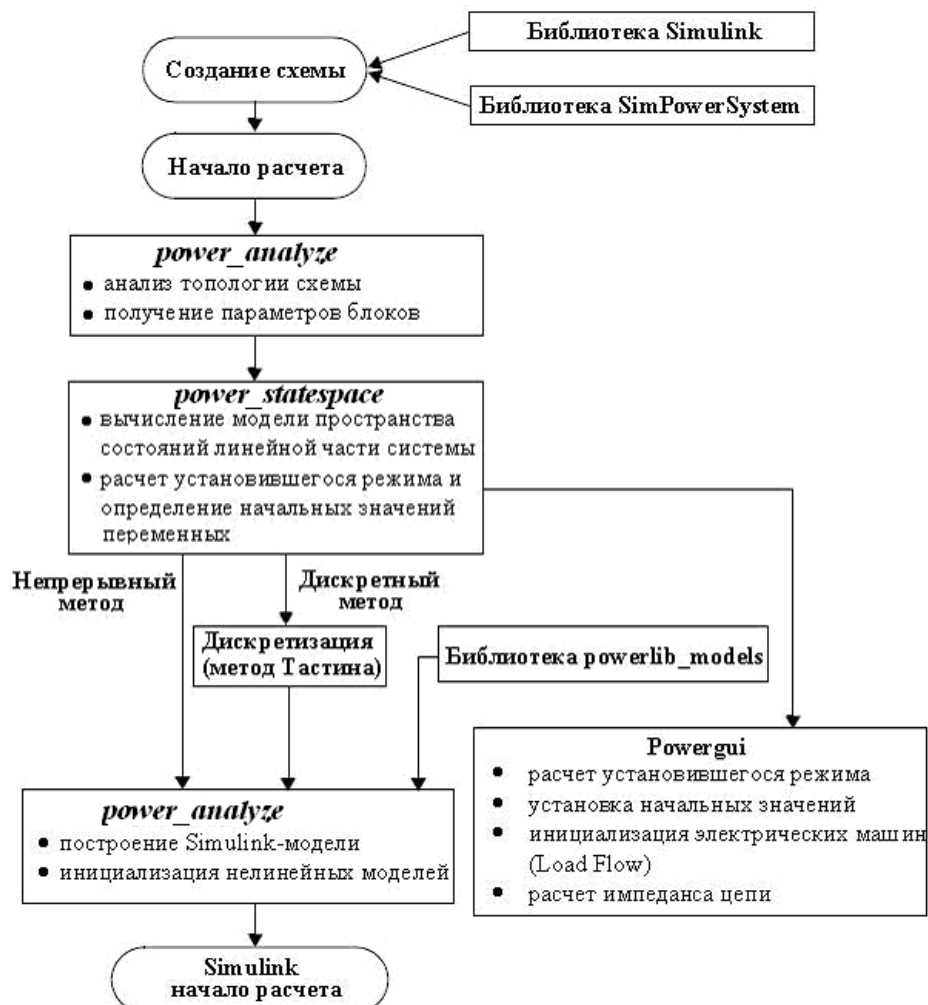





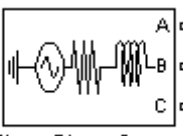
Рис. 95. Алгоритм расчета SPS-модели

В эквивалентной **Simulink**-модели используется блок **State-Space** или блок **S-function** для моделирования линейной части системы. Для моделиро-

вания **SimPowerSystem**-блоков используются их прототипы из библиотеки **powerlib_models**. Для моделирования источников энергии используются блоки источников **Soures** библиотеки **Simulink**.

2.9. Основные блоки библиотеки SimPowerSystems

Блоки источников электрической энергии используются для моделирования различных электрических сигналов (токов и напряжений). Эти блоки являются идеальными источниками (если это не оговорено заранее), т.е. их собственное сопротивление равно нулю.

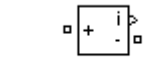
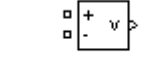



Electrical Sources – источники электрической энергии	
 DC Voltage Source	Идеальный источник постоянного напряжения – вырабатывает постоянное по уровню напряжение.
 AC Current Source	Идеальный источник переменного тока – вырабатывает синусоидальный ток с постоянной амплитудой, частотой и фазой.
 AC Voltage Source	Идеальный источник переменного напряжения – вырабатывает синусоидальное напряжение с постоянной амплитудой, частотой и фазой.
 Controlled Current Source	Управляемый источник тока – вырабатывает ток в соответствии с сигналом управления.
 Controlled Voltage Source	Управляемый источник напряжения – вырабатывает напряжение в соответствии с сигналом управления.
 Three-Phase Source	Трехфазный источник напряжения – вырабатывает трехфазную систему напряжений. Можно выбирать способы соединения фаз источника и задать полное сопротивление источника.
 Three-Phase Programmable Voltage Source	Трехфазный программируемый источник напряжения – вырабатывает трехфазную систему напряжений с программируемыми во времени изменениями амплитуды, фазы, частоты, а также гармонического состава.

Практически у всех блоков источников электрической энергии есть общие параметры:


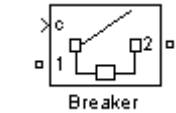
Measurments (измеряемые переменные) – этот параметр позволяет выбрать, передаваемые в блок **Multimeter**, переменные, которые затем можно увидеть с помощью блока **Scope**. Значения параметра выбираются из списка:



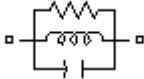
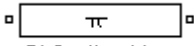
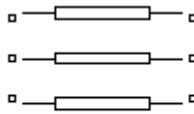
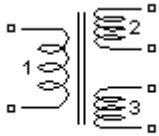
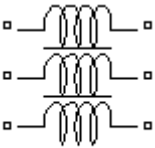
- **None** – нет переменных для отображения,
- **Voltage (Current)** – выходное напряжение (ток) источника.

Sample time (шаг дискретизации) – этот параметр задает шаг дискретизации по времени выходного сигнала источника при создании дискретных моделей.

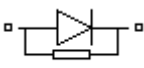
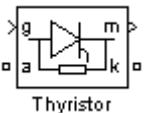
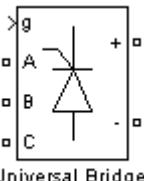
Measurements – измерительные и контрольные устройства	
 Current Measurement	Амперметр – выполняет измерение мгновенного значения тока, протекающего через соединительную линию (провод). Выходным сигналом блока является обычный сигнал Simulink , который может использоваться любым Simulink -блоком
 Voltage Measurement	Вольтметр – выполняет измерение мгновенного значения напряжения между двумя узлами схемы. Выходным сигналом блока является обычный сигнал Simulink , который может использоваться любым Simulink -блоком.
 Impedance Measurement	Измеритель полного сопротивления (импеданса) – выполняет измерение зависимости полного сопротивления участка электрической цепи от частоты.
 Three-Phase V-I Measurement	Трехфазный измеритель – выполняет измерение токов и напряжений в трехфазных цепях.
 Multimeter	Мультиметр – выполняет измерение токов и напряжений блоков библиотеки SimPowerSystem для которых в их окне диалога установлен параметр Measurements (измеряемые переменные).

Рассмотрим только некоторые блоки библиотеки элементов. Для всех элементов в библиотеке существуют аналоги, связанные с трехфазными цепями, которые мы рассматривать не будем.

Elements – электротехнические элементы	
 node 10	Заземление или нейтраль – задает «землю» на электрической схеме или одинаковые потенциалы для блоков с одинаковыми номерами узлов.
 Breaker	Выключатель переменного тока – моделирует устройство включения и выключения переменного тока. Выключатель может управляться внешним входным сигналом или от встроенного таймера. Включение

	<p>устройства выполняется единичным управляющим сигналом. Команда на выключение дается нулевым уровнем сигнала, при этом выключение устройства осуществляется при уменьшении тока до нуля. Устройство имеет встроенную искрогасящую RC-цепь, включенную параллельно контактам выключателя.</p>
 Connection Port	<p>Соединительный порт – блок, с помощью которого происходит связь подсистемы с основной моделью.</p>
 Series RLC Branch	<p>Последовательная RLC-цепь – моделирует последовательное включение резистора, индуктивности и конденсатора.</p>
 Parallel RLC Branch	<p>Параллельная RLC-цепь – моделирует параллельное включение резистора, индуктивности и конденсатора.</p>
 Pi Section Line	<p>Линия электропередачи с сосредоточенными параметрами – моделирует однофазную линию электропередачи с сосредоточенными параметрами. В реальной линии электропередачи сопротивления, индуктивность, и емкость равномерно распределены вдоль линии. Приближенная модель линии может содержать от одной до нескольких идентичных секций с сосредоточенными параметрами. Число секций зависит от частотного диапазона, который необходимо охватить при моделировании.</p>
 Distributed Parameters Line	<p>Линия электропередачи с распределенными параметрами - моделирует многофазную линию электропередачи с распределенными параметрами.</p>
 Linear Transformer	<p>Линейный трансформатор – моделирует трех или двух обмоточный однофазный трансформатор. Нелинейность характеристики намагничивания материала сердечника не учитывается.</p>
 Mutual Inductance	<p>Взаимная индуктивность – этот блок предназначен для моделирования катушек или проводников имеющих магнитную связь. Он позволяет моделировать три или два магнитно-связанных элемента.</p>

Силовая электроника – направление техники, в котором разрабатываются устройства питания электронной аппаратуры и электродвигателей. В библиотеку **Power Electronics** входят модели таких полупроводниковых элементов как диод, тиристор, транзистор, ключ и мосты, собранные из этих элементов.

Power Electronics – элементы силовой электроники.	
 <p>Diode</p>	Силовой диод – моделирует полупроводниковый силовой диод.
 <p>Thyristor</p>	Тиристор – моделирует тиристор. В библиотеке SimPowerSystem имеется две модели тиристора: Thyristor (упрощенная модель) и Detailed Thyristor (уточненная модель).
 <p>Universal Bridge</p>	Универсальный мост – моделирует универсальный мост. Модель позволяет выбирать количество плеч моста (от 1 до 3), вид полупроводниковых приборов (диоды, тиристоры, идеальные ключи, а также полностью управляемые тиристоры, IGBT и MOSFET транзисторы, шунтированные обратными диодами). В модели можно также выбрать вид зажимов A, B и C (входные или выходные).

2.10. Powergui – графический интерфейс пользователя

Анализ электрических цепей удобно проводить с помощью блока **Powergui** – графического интерфейса пользователя. Этот блок расположен отдельно в библиотеке **SimPowerSystems**, и его можно «перетащить» в окно **Simulink**-модели, как обычный блок. Но в этом нет необходимости, т.к. блок **Powergui** появляется в окне схемы, содержащей блоки **SPS**, при первом запуске. Этот блок не просто удобен, он необходим для работы **SPS** – модели, т.к. он используется для хранения эквивалентной **Simulink**-модели в пространстве состояний. Отметим некоторые особенности блока **Powergui**:

- он может находиться в любой части модели (в том числе и в окне подсистемы);
- в одной модели может находиться только один блок **Powergui**;
- блок **Powergui** должен иметь свое уникальное имя, автоматически блоку присваивается имя модели.

После двойного щелчка по блоку появляется окно блока **Powergui** (Рис. 96). Как видно из рисунка, блок **Powergui** обеспечивает решение следующих задач:

- В верхней части окна осуществляется выбор метода расчета **Simulation type**:
 - расчет схемы методом комплексных амплитуд (при этом необходимо задать частоту источников в графе **Frequency**). Этот метод используется для расчета линейных радиотехнических цепей, со-

держативные реактивные элементы, в установившемся режиме при гармонических входных сигналах. Для всех реактивных элементов определяется их комплексный импеданс, а токи и напряжения рассматриваются в виде комплексных амплитуд.

- дискретизация электрической модели (при этом необходимо задать шаг дискретизации в графе **Sample time**). Если выбран этот метод, то рассчитывается дискретная модель с постоянным периодом квантования. Рекомендуется использовать для больших сложных электрических схем. Тогда решение ищется с постоянным шагом по времени, за счёт этого значительно может быть сокращено время моделирования.
 - расчет «непрерывной» электрической модели. В этом случае рассчитывается модель с переменным шагом по времени.
- Ниже (**Analysis tools**) расположены кнопки, с помощью которых можно открыть соответствующие инструменты **SimPowerSystems**:
 - расчет установившегося режима для токов и напряжений;
 - задание начальных условий (начальные напряжения для конденсаторов и токи для катушек);
 - получение информации (инициализация) о трехфазных схемах, содержащих электрические машины из библиотеки **Machines**, расчет потокораспределения нагрузки **load flow**;
 - анализ схемы с помощью инструмента **Simulink LTI-Viewer**;
 - определение полного сопротивления (импеданса) цепи. Открывается окно, в котором показан график зависимости импеданса цепи от частоты;
 - выполнение быстрого преобразования Фурье **FFT**;
 - создание отчета;
 - создание файла характеристик намагничивания для модели нелинейного трансформатора;
 - расчет *RLC* параметров длинной линии.

При загрузке **LTI-Viewer** появится окно **Powergui link to LTI View model**, в котором можно выбрать вход и выход исследуемой системы (Рис. 97).

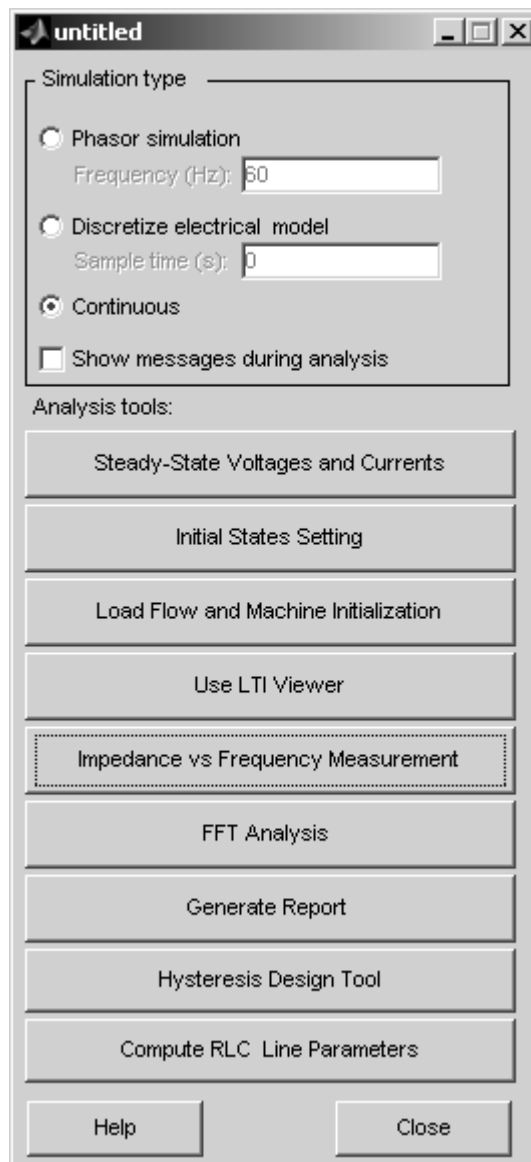


Рис. 96. Окно блока Powergui

Для просмотра результатов линейного анализа необходимо нажать кнопку **Open new LTI Viewer** и выбрать вид графика в окне **LTI Viewer**. Дальнейшая работа с этим инструментом уже описывалась в п. 2.7.

С помощью блока **Powergui** также можно установить начальные значения в цепи. При нажатии на кнопку **Initial States Setting** открывается окно, в котором отображаются начальные значения переменных. Эти значения можно изменять. Новые значения используются при расчете переходных процессов.

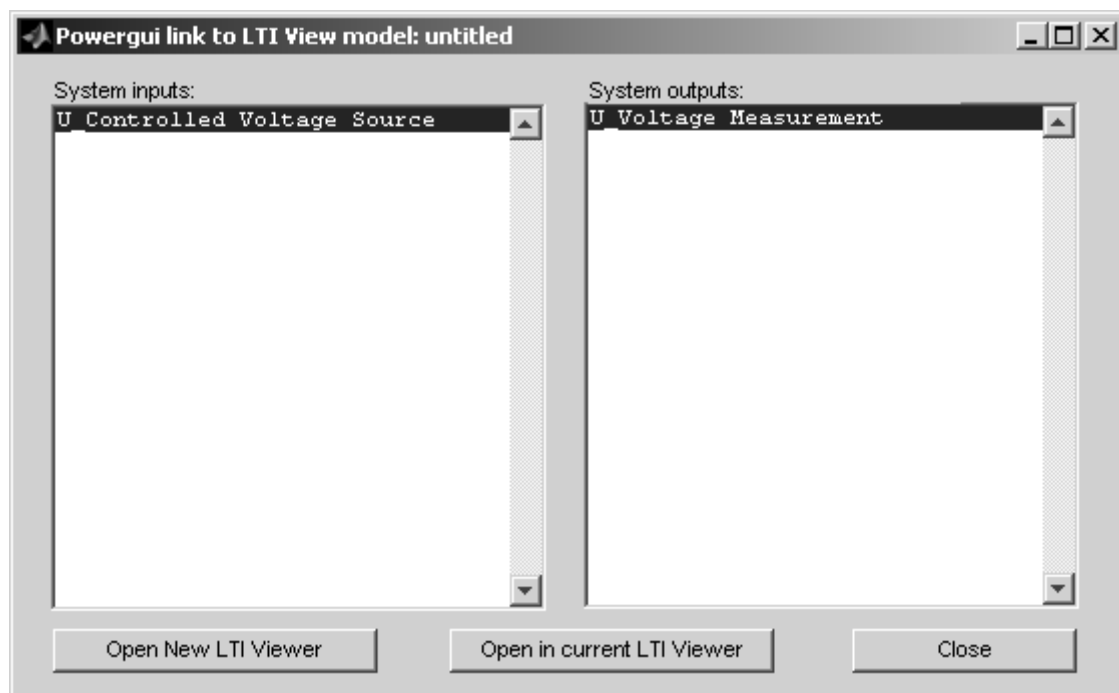


Рис. 97. Окно Powergui link to LTI View model

2.11. Примеры построения SPS-моделей

Рассмотрим несколько задач, для решения которых можно использовать **SimPowerSystems**.

2.11.1. Зарядка конденсатора

Требуется с помощью **SimPowerSystems** построить модель электрической схемы, состоящей из источника постоянного напряжения 200 В, сопротивления $R = 1\text{ кОм}$ и конденсатора $C = 1\text{ мФ}$ (эта модель в приложении называется **capacitor.mdl**). В начальный момент времени конденсатор не заряжен. С помощью амперметра нужно проследить, как будет меняться электрический ток в этой цепи.

Для решения этой задачи, откроем рабочее окно **Simulink** и «перетащим» туда мышкой блоки «Идеальный источник постоянного напряжения», «Последовательная RLC -цепь», «Амперметр» из библиотеки **SimPowerSystems** и блок «Осциллограф» из библиотеки **Simulink**. Соединим эти блоки, как показано на Рис. 98. Обратим внимание, что соединительные линии, изображающие провода электрической схемы, не имеют стрелок. Линия со стрелкой соединяет амперметр и осциллограф и показывает направление передачи сигнала.

Рассмотрим подробнее блоки, входящие в схему. Для источника постоянного напряжения в окне параметров нужно задать величину сигнала 200 В (Рис. 99).

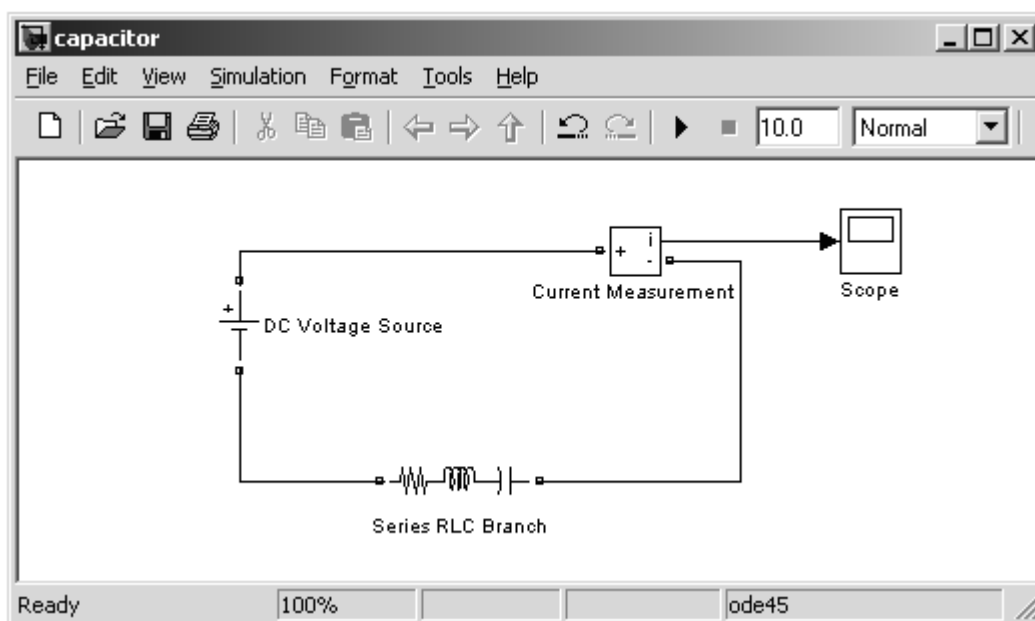


Рис. 98. Пример моделирования в SimPowerSystems

Блоки RLC -цепочек (параллельной и последовательной) рассчитаны на различный состав элементов. Окно параметров блока параллельной RLC -цепочки изображено на Рис. 100.

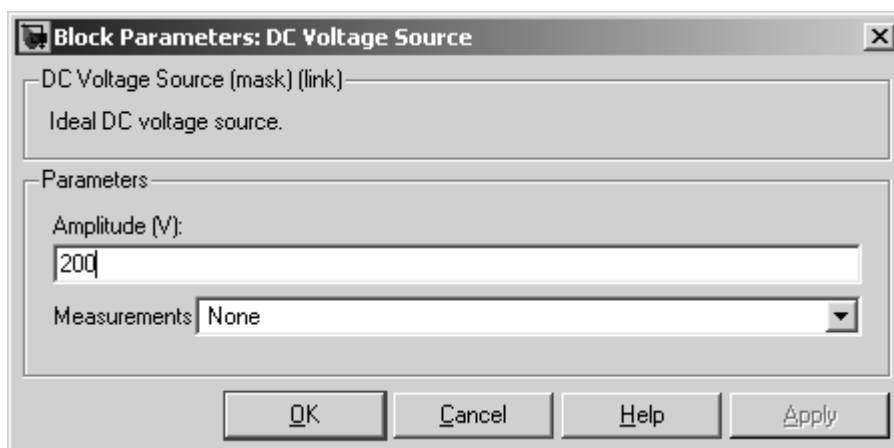


Рис. 99. Окно параметров блока источника постоянного напряжения

В выпадающем списке параметра **Branch Type** (Рис. 101) можно выбрать число элементов цепочки. Для нашей задачи нужно оставить только резистор и конденсатор. После выбора определенного типа цепочки, изображение блока (как и окно параметров) изменится, иконка блока будет содержать только выбранные элементы. Если требуется включить несколько одинаковых элементов, например, последовательные сопротивления, то нужно использовать несколько блоков RLC -цепочек с соответствующими параметрами. В окнах **Resistance** и **Capacitance** вводим заданные значения сопротивления в омах и емкости в фарадах. Пункт **Set the initial capacitor voltage** (установить начальное значение напряжения на конденсаторе) активизирует

учет переходных процессов при решении, поэтому должен быть выбран, даже если начальное значение напряжения на конденсаторе равно нулю.

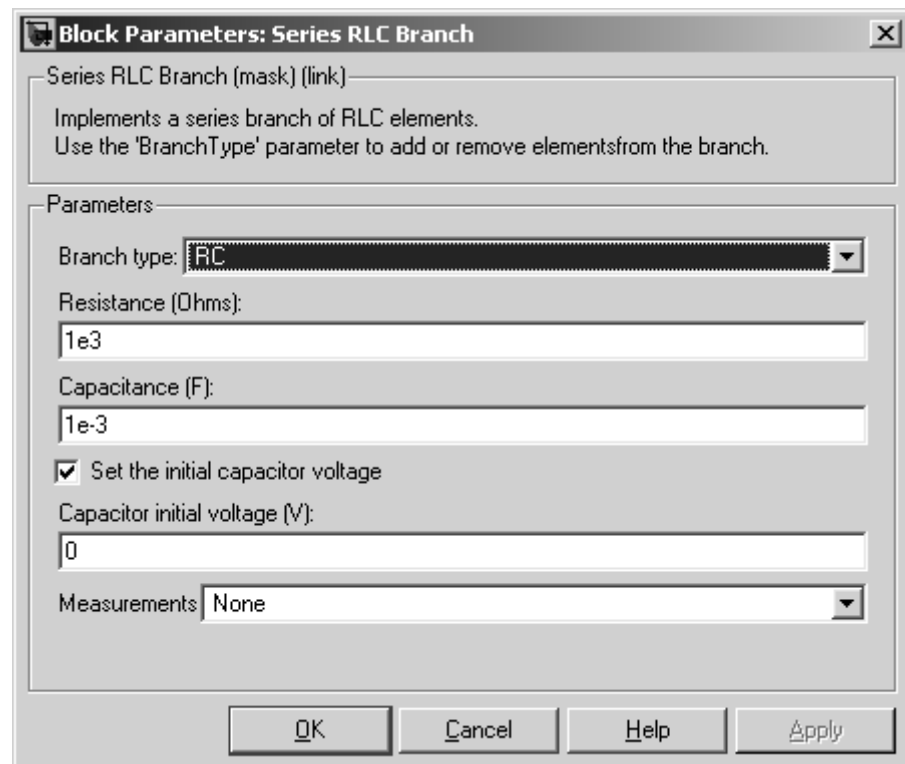


Рис. 100. Окно параметров блока параллельной RLC-цепочки

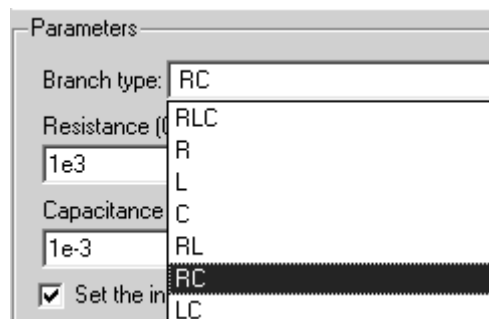


Рис. 101. Выпадающий список параметра Branch Type

В окне параметров блока амперметр в нашем случае ничего изменить нельзя (Рис. 102).

Единственный параметр **Output signal** (вид выходного сигнала блока) доступен только, если с помощью блока **Powergui** установлен режим расчета на переменном токе (**Phasor simulation**). В этом случае значение параметра выбирается из списка:

- **Magnitude** – амплитуда (скалярный сигнал);
- **Complex** – комплексный сигнал;

- **Real-Imag** – вектор, состоящий из двух элементов - действительная и мнимая составляющие сигнала;
- **Magnitude-Angle** – вектор, состоящий из двух элементов - амплитуда и аргумент сигнала.

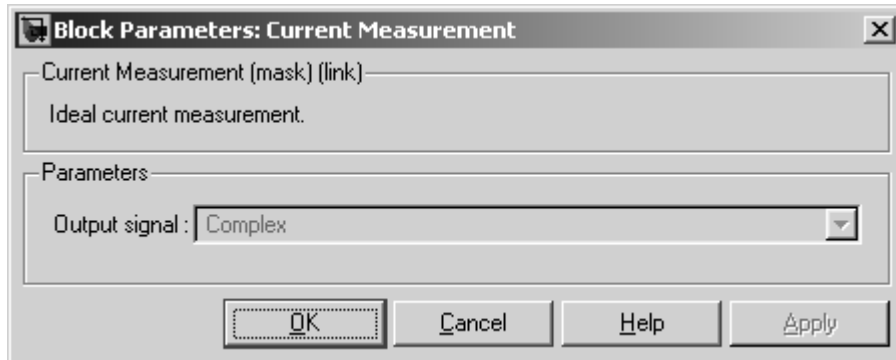


Рис. 102. Окно параметров блока амперметр

После проведения моделирования (команда **Start simulation**) в рабочем окне появляется дополнительный блок **Powergui**. Окончательно схема будет выглядеть, как показано на Рис. 103.

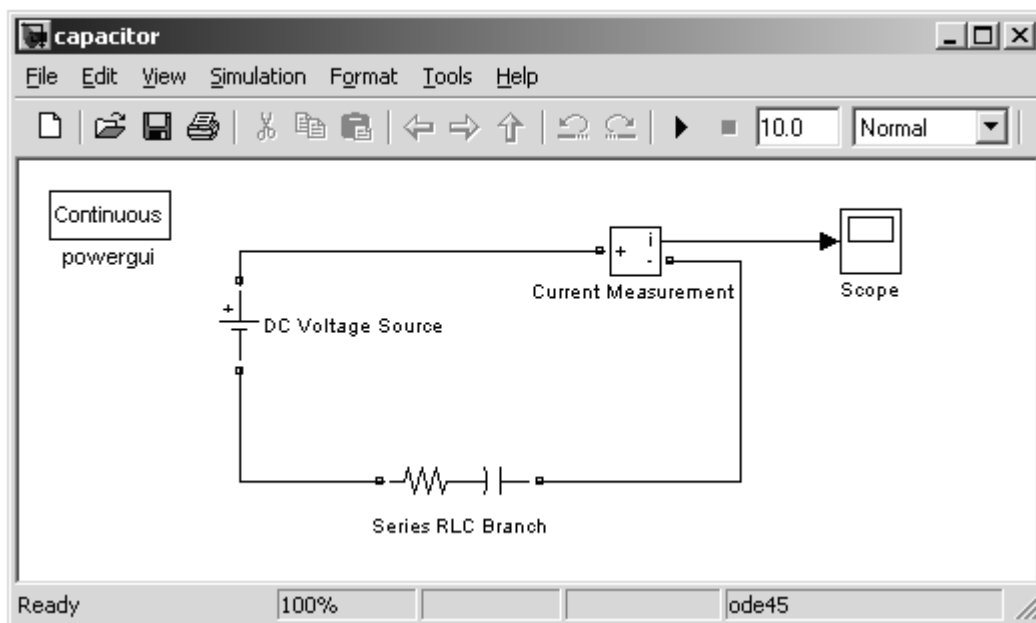


Рис. 103. Окончательная схема моделирования зарядки конденсатора

Известно, что процесс зарядки конденсатора через сопротивление описывается экспоненциальной функцией. Для рассматриваемой задачи легко получить аналитическую зависимость силы тока от времени:

$$I = \frac{E_0}{R} \exp\left(-\frac{t}{\tau}\right),$$

где E_0 – напряжение источника, а постоянная времени $\tau = RC$ (в нашем случае $\tau = 1$ с). Изменение электрического тока в цепи со временем показано на Рис. 104.

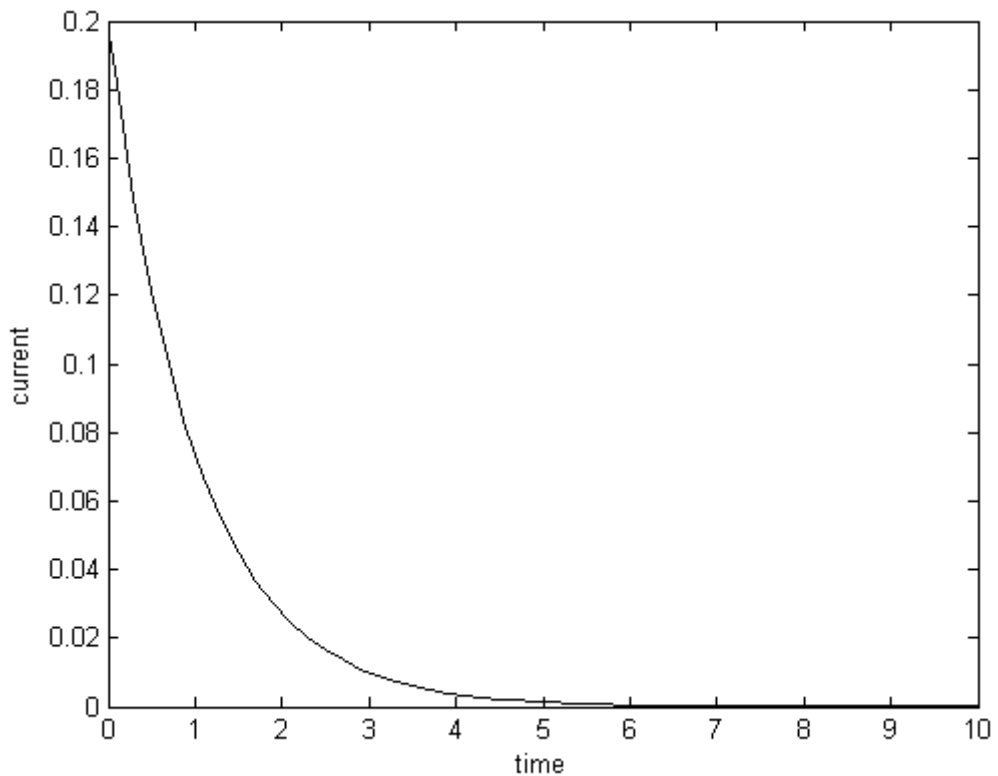


Рис. 104. Изменение электрического тока в цепи со временем

Можно проверить, совпадают ли результаты моделирования с теоретической зависимостью, например, собрав в рабочем окне дополнительную схему из блоков **Simulink** (Рис. 106). Здесь, для того чтобы вывести обе зависимости на один график, используется блок «мультиплексор» или «смеситель» **mux** из библиотеки **Signal Routing**. Его пентаграмма изображена на Рис. 105.



Рис. 105. Иконка блока «мультиплексор» mux

Этот блок объединяет несколько (количество можно задавать в окне параметров блока) сигналов в вектор или матрицу.

Результат работы этой схемы не отличается от результата, приведенного на Рис. 104.

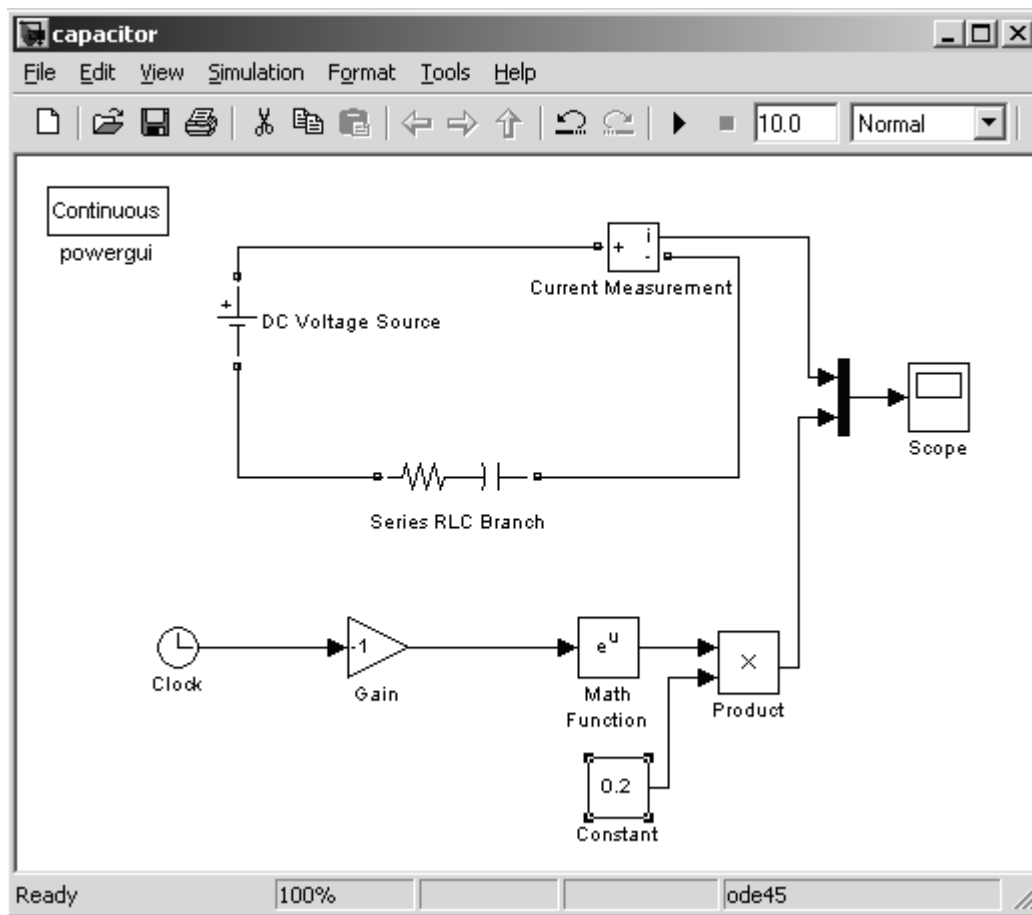


Рис. 106. Схемы SimPowerSystems и Simulink в одном рабочем окне

2.11.2. RC-фильтр низких частот

Рассмотрим схему RC-фильтра низких частот (Рис. 41). В приложении эта схема называется **low_filter_RC.mdl**. В параграфе 2.2.4. мы уже применяли для описания этой схемы передаточную функцию. Пакет **SimPowerSystems** позволяет собирать электрическую схему из блоков, не вникая в математическое описание ее работы. Для построения схемы, «перетащим» в рабочее окно **Simulink** два блока **Series RLC Branch**. Один из этих блоков будет моделировать резистор, а другой – конденсатор. В окне параметров **Branch Type** для каждого из этих блоков выберем соответственно R и C . Свойства цепочки определяются постоянной времени τ . В рассматриваемом ранее примере (в параграфе 2.2.4.) с передаточной функцией была выбрана постоянная времени цепочки $\tau = 0,1$ с. Чтобы иметь возможность сравнивать результаты моделирования в **Simulink** и в **SimPowerSystems** оставим прежнее значение τ . Для этого выберем сопротивление резистора 10^3 Ом и емкость конденсатора 10^{-4} Ф так, чтобы $\tau = RC = 0,1$ с. Чтобы схема лучше читалась, повернем блок для конденсатора на 90 градусов. Для этого нужно выделить блок, выбрать команды **Format/Rotate block**. Добавим в схему два блока из

библиотеки **Simulink**: источник сигнала – генератор прямоугольных импульсов и приемник сигнала – осциллограф (Рис. 107).

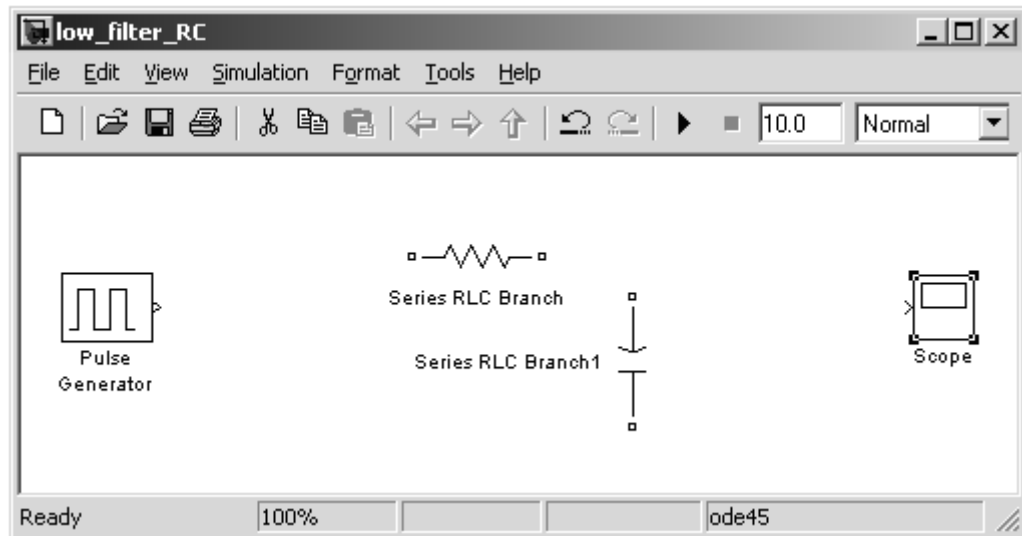


Рис. 107. RC-фильтр низких частот: начало построения схемы

Как уже обсуждалось раньше, блоки **Simulink** нельзя непосредственно соединять с блоками **SimPowerSystems**, поэтому добавим в схему еще блоки управляемого источника напряжения и приемник напряжения. После соединения блоков, установки периода равного 3 с для блока прямоугольных импульсов и проведения моделирования схема будет выглядеть, как показано на Рис. 108.

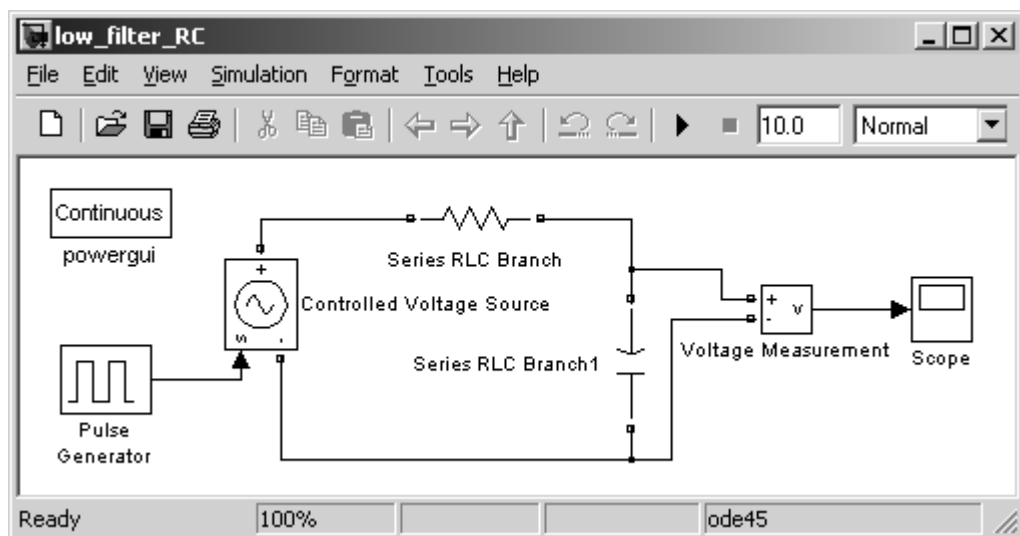


Рис. 108. RC-фильтр низких частот: окончательная схема

Результат моделирования совпадает с результатом, приведенным на Рис. 45. В **Powergui** можно получить АЧХ и ФЧХ этой цепочки (Рис. 109).

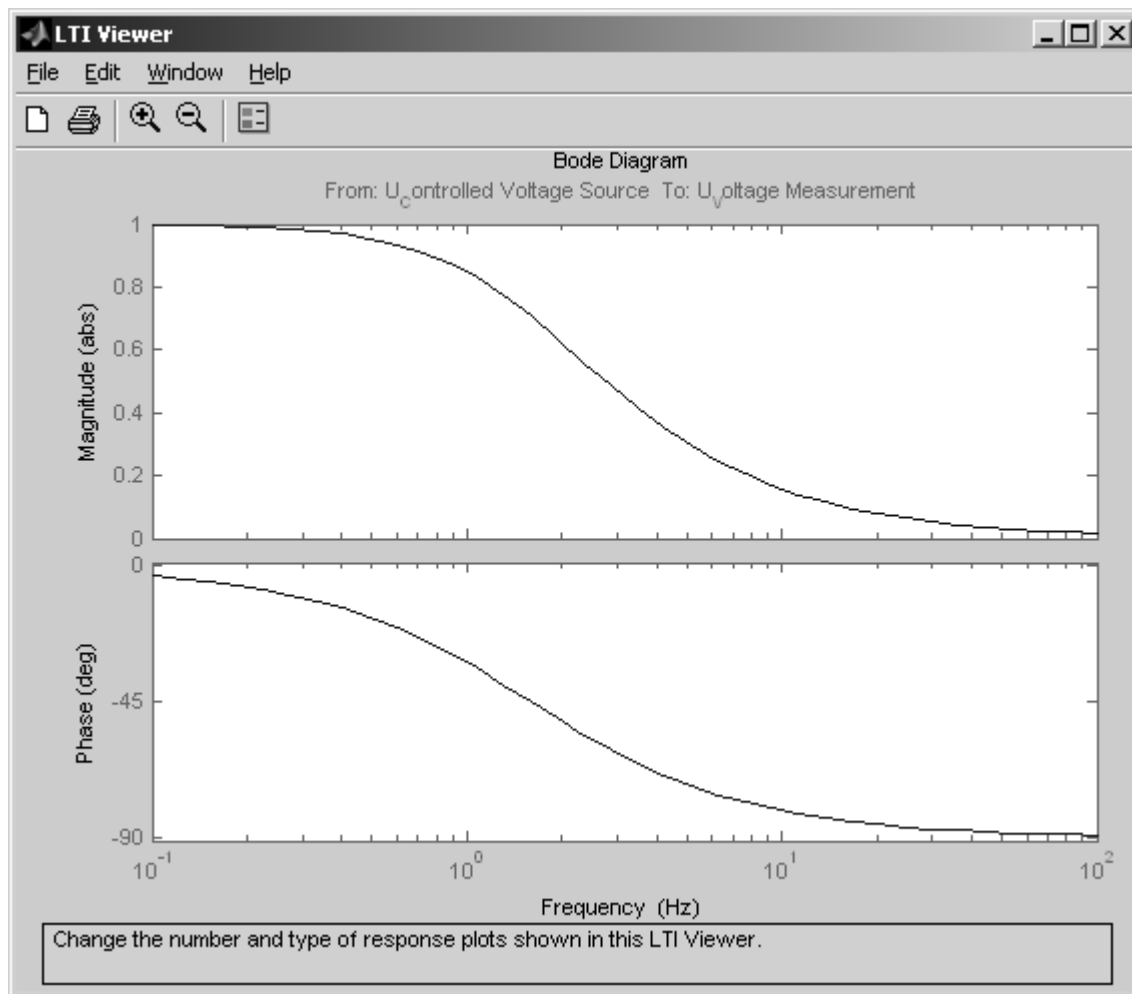


Рис. 109. АЧХ и ФЧХ фильтра низких частот

2.11.3. Цепочка Вина

Рассмотрим симметричную цепочку Вина. Она состоит из последовательно и параллельно включенных конденсаторов и резисторов (Рис. 110). Для симметричной цепочки значения емкостей и сопротивлений выбираются одинаковые.

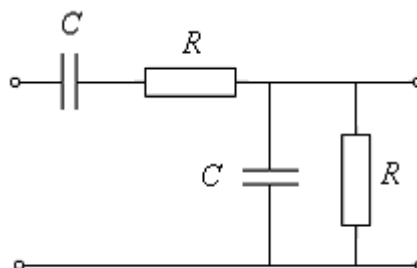


Рис. 110. Симметричная цепочка Вина

На Рис. 111 изображена **Simulink**-схема, моделирующая цепочку Вина (в приложении она называется **vina.mdl**). Источником сигнала на схеме является источник переменного напряжения. Выходное напряжение с помощью блока **Voltage Measurement** регистрируется осциллографом. Цепочка Вина является избирательным фильтром. Легко получить выражение для коэффициента передачи цепочки Вина. Учитывая, что входной импеданс этой цепочки равен

$$Z_{\text{вх}} = R + \frac{1}{i\omega C} + \frac{\frac{1}{i\omega C} R}{\frac{1}{i\omega C} + R},$$

а выходной –

$$Z_{\text{вых}} = \frac{\frac{1}{i\omega C} R}{\frac{1}{i\omega C} + R}.$$

Вводя обозначение $\omega_0 = \frac{1}{RC}$, получаем

$$\dot{K} = \frac{1}{3 + i \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right)}. \quad (10)$$

Отсюда модуль коэффициента передачи будет

$$|\dot{K}| = \frac{1}{\sqrt{9 + \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right)^2}} \quad (11)$$

и сдвиг фаз

$$\Delta\varphi = -\arctg \left(\frac{1}{3} \left(\frac{\omega}{\omega_0} - \frac{\omega_0}{\omega} \right) \right). \quad (12)$$

Из этих выражений видно, что максимальный коэффициент передачи цепочки Вина равен 1/3 на частоте $f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi RC}$, а сдвиг фаз на этой частоте равен нулю.

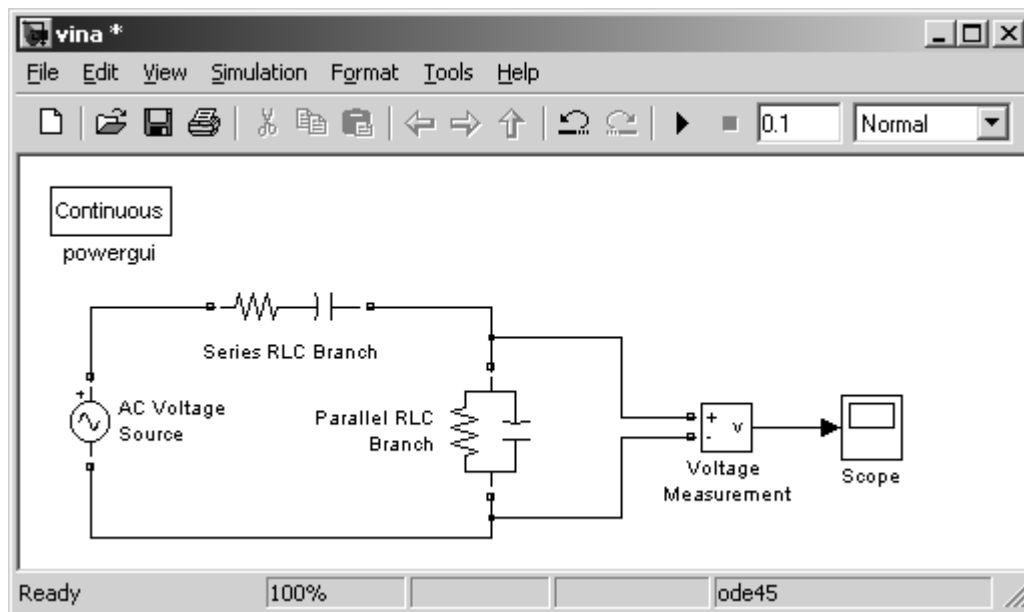


Рис. 111. Схема с цепочкой Вина

АЧХ и ФЧХ симметричной цепочки Вина показаны на Рис. 112 для значений $R=10^3/2\pi$ Ом и $C=10^{-6}$ Ф.

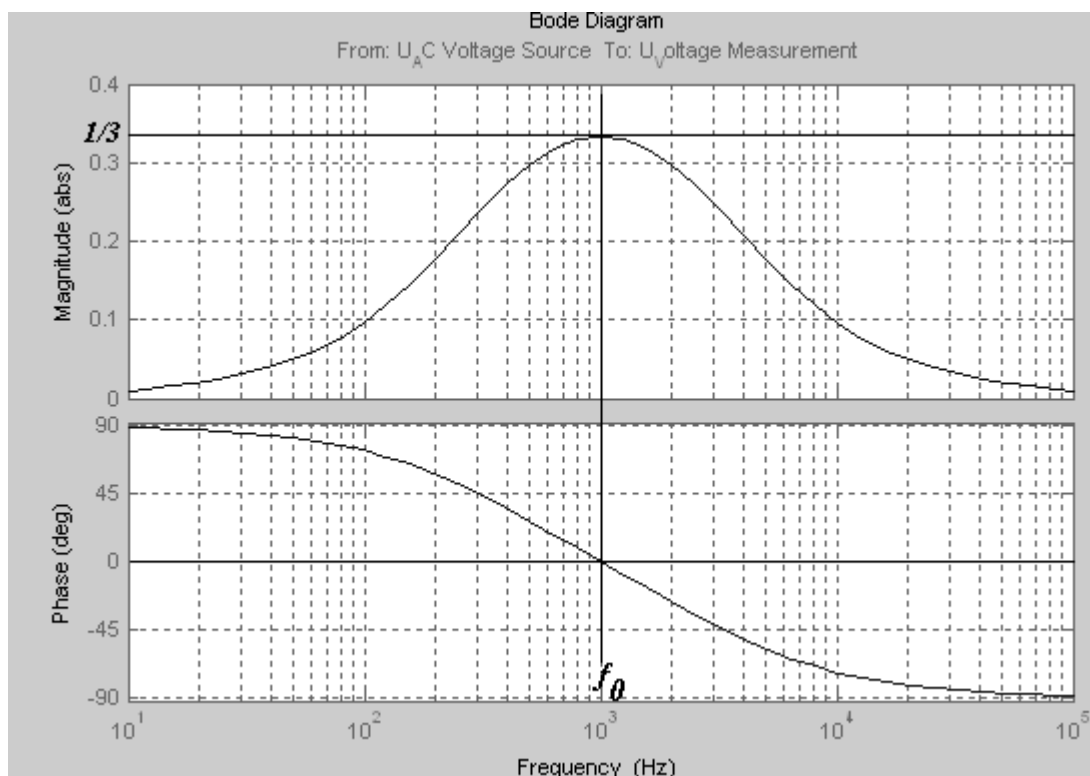


Рис. 112. АЧХ и ФЧХ симметричной цепочки Вина

Видно, что в данном случае максимум будет на частоте 10^3 Гц.

2.11.4. Диод

Рассмотрим моделирование нелинейного элемента на примере полупроводникового диода. Диод обладает резко выраженной односторонней проводимостью электрического тока. На Рис. 113 показано условное обозначение диода на электрических схемах.

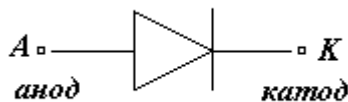


Рис. 113. Изображение диода на электрических схемах

Статическая вольт-амперная характеристика модели диода показана на Рис. 114. Здесь приводятся обозначения (ток в направлении анод-катод I_{ak} , напряжение на диоде V_{ak} , запирающее напряжение V_f , сопротивление диода в открытом состоянии R_{on}), используемые в справочной системе **SimPowerSystems**.



Рис. 114. Вольт-амперная характеристика диода

Модель диода в **SimPowerSystems** состоит из последовательно включенных резистора R_{on} , индуктивности L_{on} , источника постоянного напряжения V_f и ключа SW (Рис. 115). Блок логики управляет работой ключа. При положительном напряжении на диоде ($V_{ak} - V_f$) происходит замыкание ключа и через прибор начинает протекать ток. Размыкание ключа (выключение диода) выполняется при снижении тока I_{ak} , протекающего через диод, до нуля.

Дополнительно к этой схеме в **SimPowerSystems** предусмотрено включение параллельно самому диоду последовательной RC-цепочки (с параметрами R_s и C_s), выполняющей демпфирующие функции.

Окно задания параметров блока показано на Рис. 116. Обозначения соответствуют Рис. 115.

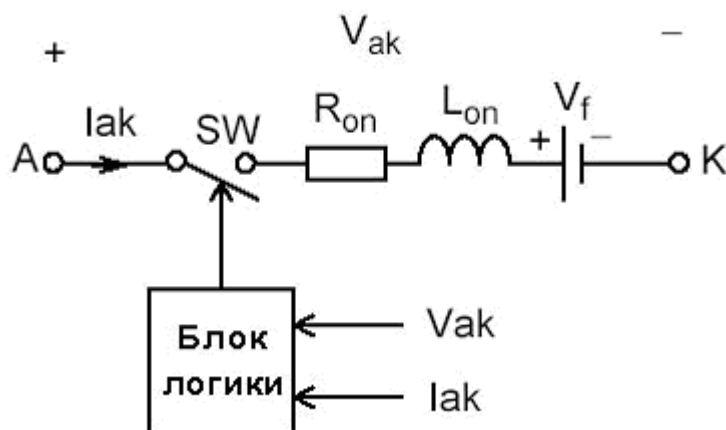


Рис. 115. Модель диода в SPS

На выходном порту блока, обозначенном **m**, формируется векторный **Simulink**-сигнал из двух элементов. Первый элемент – анодный ток диода, второй – напряжение анод-катод диода. Эта опция включается в окне параметром **Show measurement port**.

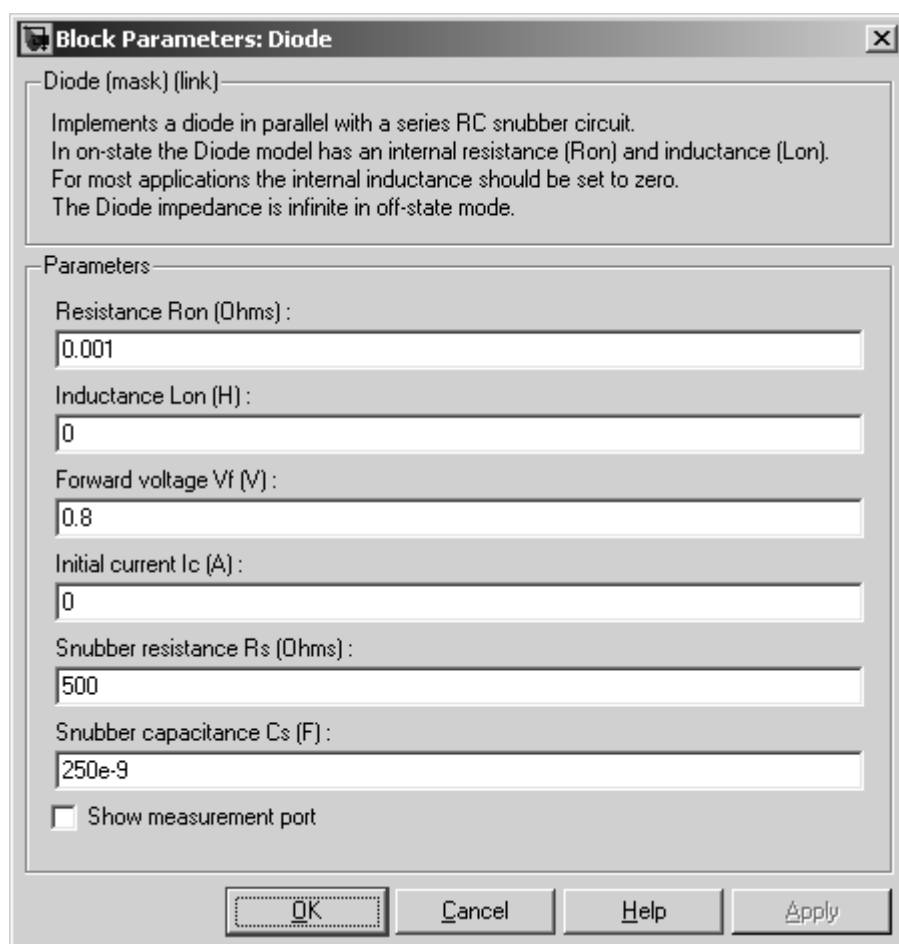
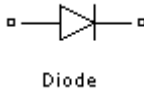
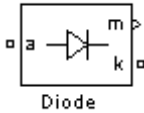
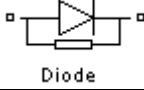
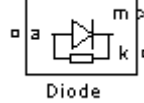


Рис. 116. Окно параметров блока диод

В зависимости от того, какие параметры были выбраны, пиктограмма блока может иметь разный вид (см. таблицу).

пиктограмма	Значения сопротивления R_s и емкости C_s демпфирующей цепи	Включена опция Show measurement port
 Diode	$R_s = 0$ $C_s = 0$	нет
 Diode	$R_s = 0$ $C_s = 0$	да
 Diode	заданы не равными нулю	нет
 Diode	заданы не равными нулю	да

На Рис. 117 изображено рабочее окно **Simulink** с электрической схемой (в приложении модель называется **diode.mdl**). Источником служит источник гармонического напряжения ($u = 220\sin(100\pi t)$). В схему включены резистор $R = 1\text{кОм}$ и диод. Напряжение на источнике измеряется вольтметром **Voltage Measurement** и ток в цепи – амперметром **Current Measurement**.

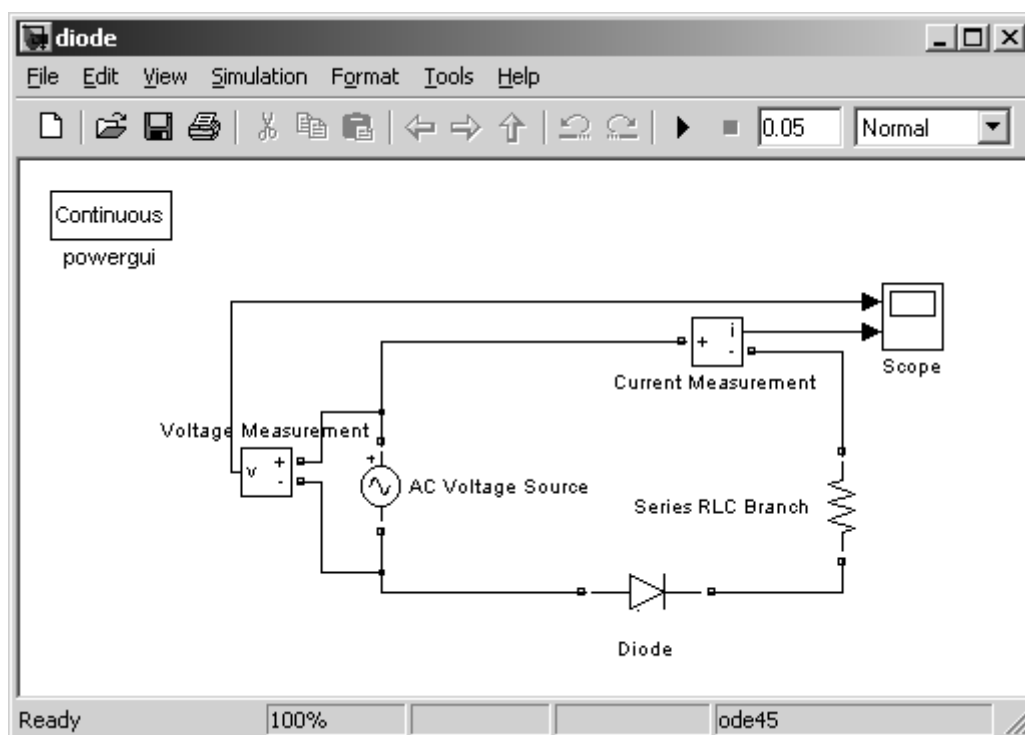


Рис. 117. Электрическая схема с диодом

Параметры диода показаны на Рис. 118. Здесь не включены демпфирующая цепь и измерительный порт диода.

Parameters	
Resistance R_{on} (Ohms) :	<input type="text" value="0.001"/>
Inductance L_{on} (H) :	<input type="text" value="0"/>
Forward voltage V_f (V) :	<input type="text" value="0.8"/>
Initial current I_c (A) :	<input type="text" value="0"/>
Snubber resistance R_s (Ohms) :	<input type="text" value="0"/>
Snubber capacitance C_s (F) :	<input type="text" value="0"/>
<input type="checkbox"/> Show measurement port	

Рис. 118. Часть окна параметров диода

На Рис. 119 показаны графики измеряемого напряжения на источнике (пунктирной линией) и тока в цепи (сплошной линией). Заметим, что для наглядности эти величины приведены на одном графике, несмотря на то, что у них разная размерность (на графике по оси ординат). Когда диод закрыт, ток в цепи не течет и на сопротивлении падение напряжения равно нулю. В открытом состоянии диода, когда через резистор течёт ток, падение напряжения на нём равно напряжению источника, так как это единственный элемент в цепи (если диод считать идеальным).

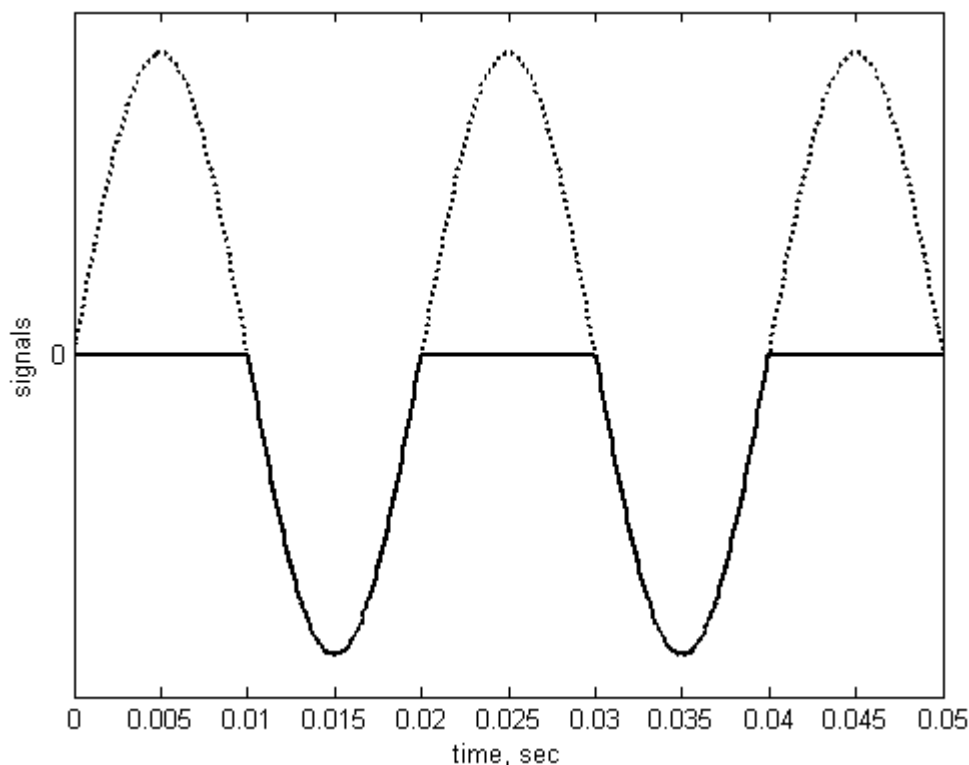


Рис. 119. Графики напряжения и тока в цепи

2.11.5. Создание электротехнических блоков пользователя

В библиотеке **SimPowerSystems** нет готового блока диода, для которого можно задавать произвольную (нелинейную) ВАХ. Также в библиотеке **SPS** может не быть других нелинейных элементов, например, нелинейных резисторов, катушек индуктивности с насыщением, новых типов электродвигателей и т.п. Но новый элемент легко создать самому пользователю на основе блоков **Simulink** и блоков **SPS**. Рассмотрим принцип создания электротехнических блоков пользователя. Общая структура модели показана на Рис. 120.

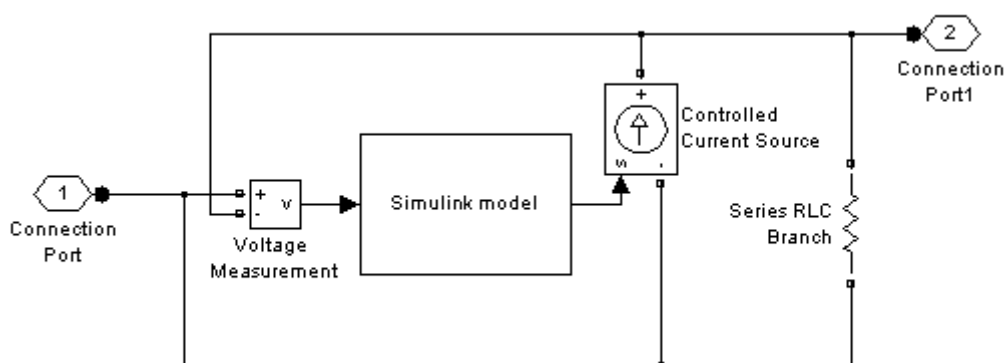


Рис. 120. Общая структура модели пользователя

На схеме к управляемому источнику тока параллельно подключен измеритель напряжения. Между выходом измерителя напряжения и входом источника тока включена **Simulink**-модель, реализующая нужную вольт-амперную характеристику устройства. Параллельно источнику также подключен развязывающий резистор. Его наличие обусловлено тем, что большое число блоков **SPS** выполнено на базе источников тока. При последовательном соединении таких блоков источники тока оказываются включенными последовательно, что недопустимо. Наличие же развязывающего резистора позволяет включать такие блоки последовательно. Величина сопротивления резистора должна выбираться достаточно большой (значительно больше, чем максимально возможное значение нелинейного сопротивления в **Simulink**-модели), чтобы его влияние на характеристики создаваемого блока было минимально (иначе может оказаться так, что это сопротивление зашунтирует нелинейный элемент). Обычно его величина для силовых электротехнических блоков выбирается в пределах 500 – 1000 Ом.

Клеммами создаваемого устройства являются входной и выходной порты модели (**Connection Port**). Таким образом, для создания электротехнического блока, пользователь, прежде всего, должен создать обычную **Simulink**-модель, входом которой является сигнал, пропорциональный приложенному к клеммам устройства напряжению. А выходом этой модели – сигнал, пропорциональный току, протекающему через устройство. Затем

нужно включить эту модель в схему, представленную на Рис. 120. При создании многофазных устройств, как правило, требуется создать модели отдельных фаз, а затем уже включить их по нужной схеме.

Создадим по этой схеме модель нелинейного резистора (в приложении модель называется **nelinR.mdl**). Пусть вольт-амперная характеристика нелинейного резистора задана выражением:

$$i = I_0 \left(\frac{u}{U_0} \right)^\alpha, \quad (13)$$

где

i и u – мгновенные значения тока и напряжения,

U_0 – пороговое значение напряжения,

I_0 – величина тока устройства при значении напряжения, равном пороговому,

α – показатель степени, определяющий нелинейность вольт-амперной характеристики резистора (обычно лежит в пределах от 0 до 50).

Схема блока нелинейного резистора показана на Рис. 121. Модель **Simulink**-резистора реализована с помощью блока **Fcn**. Численные значения параметров приняты следующими:

$U_0 = 110$ кВ,

$I_0 = 500$ А,

$\alpha = 25$.

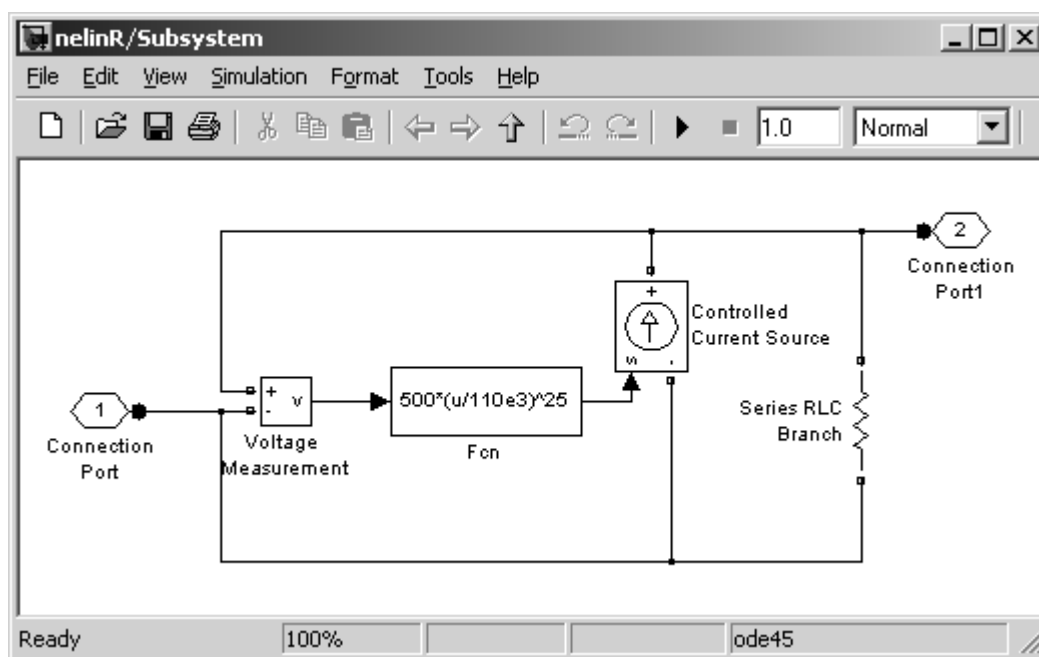


Рис. 121. Схема блока нелинейного резистора

Далее эту модель можно оформить и сохранить как подсистему и использовать в других моделях как отдельный блок. На Рис. 122 изображена модель электрической схемы с нелинейным элементом. Нелинейное сопротивление выделено в отдельную подсистему **Subsystem**. В цепь включено

нелинейное сопротивление и источник синусоидального напряжения. Измеряется напряжение на сопротивлении (**Voltage Measurement**) и ток в цепи (**Current Measurement**). Графики тока и напряжения от времени строятся с помощью осциллографа, а вольт-амперная характеристика строится с помощью графопостроителя **XY-Ggraph**.

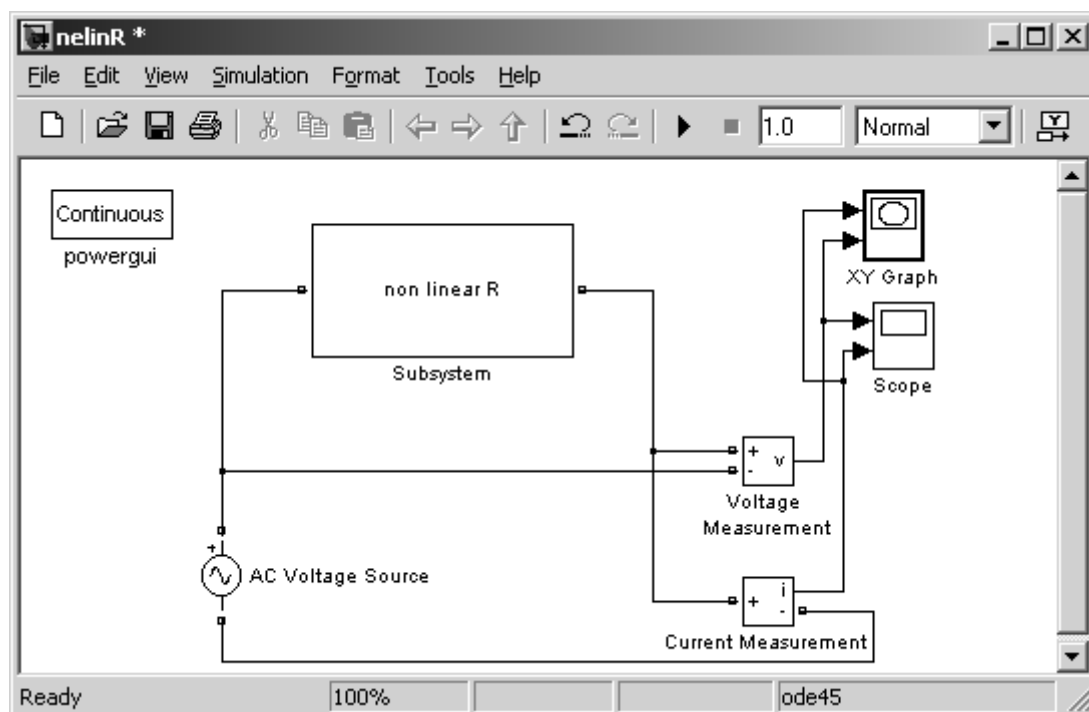


Рис. 122. Схема модели с нелинейным элементом

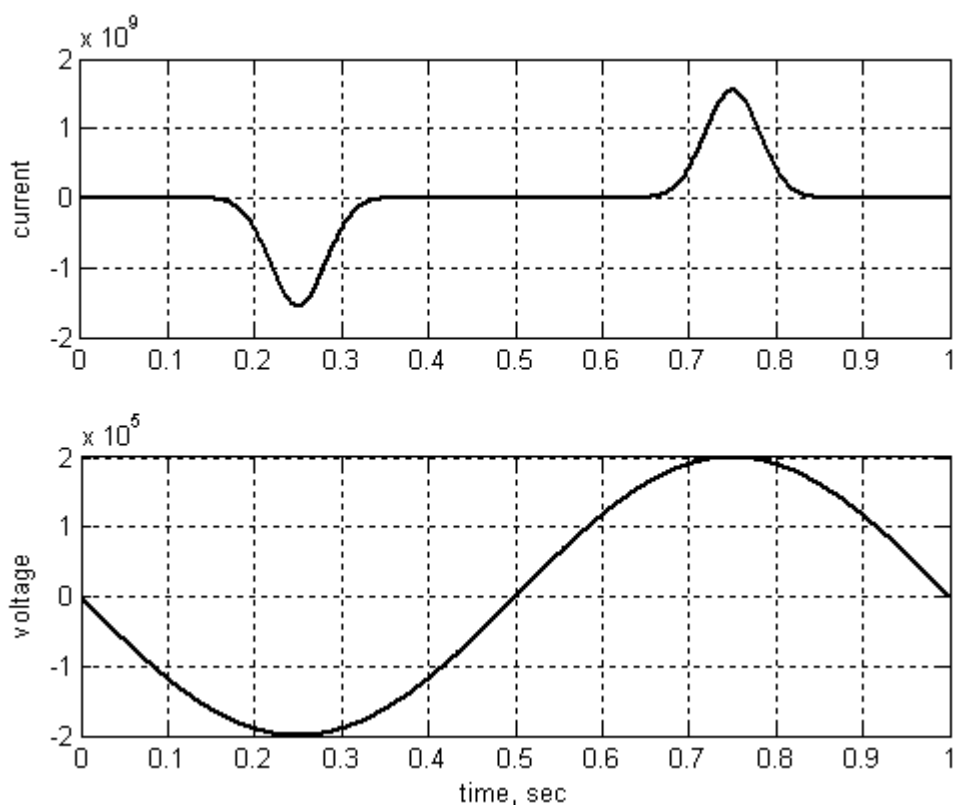


Рис. 123. Зависимость тока и напряжения от времени

Временные диаграммы работы модели показаны на Рис. 123. Вольт-амперная характеристика и диапазон шкалы графопостроителя (окно параметров блока) показаны на Рис. 124.

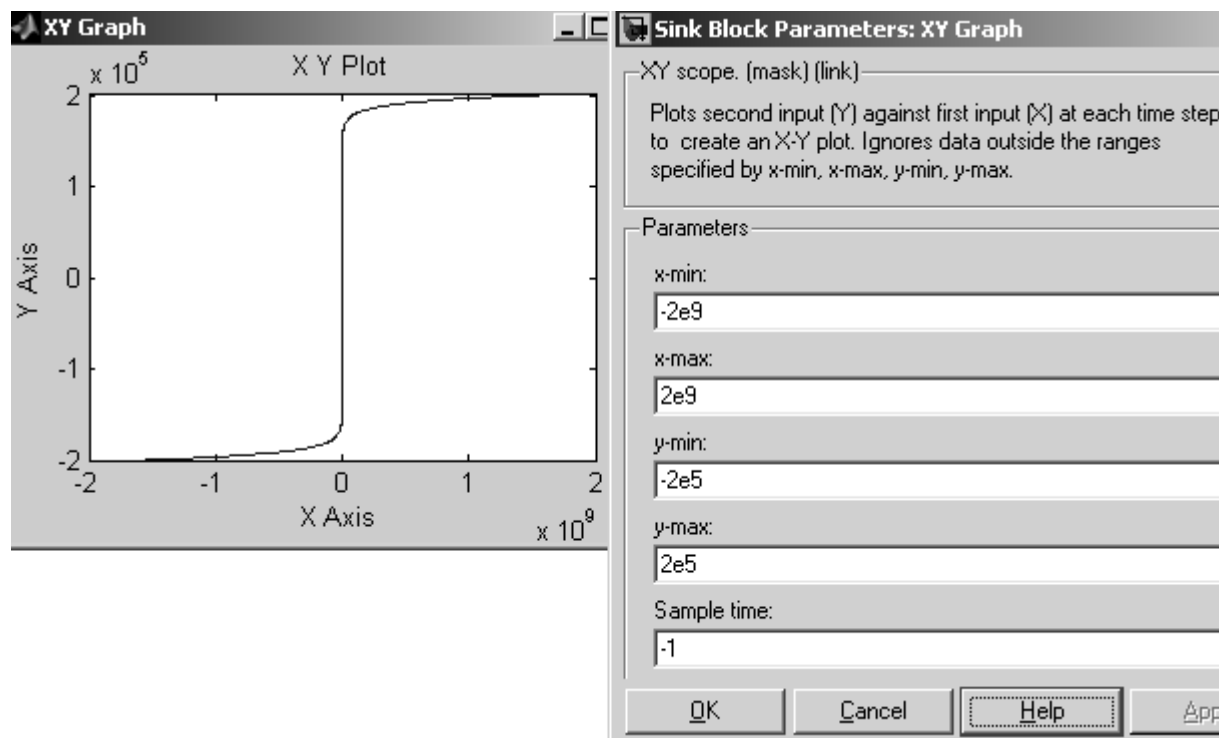


Рис. 124. Вольт-амперная характеристика

Некоторым недостатком такой модели резистора является наличие безынерционного замкнутого контура, образованного датчиком напряжения, блоком **Fcn** и управляемым источником тока (Рис. 121). **Simulink** выполняет расчет таких моделей, используя итерационную процедуру, что несколько снижает скорость расчета. Также при наличии замкнутых алгебраических контуров становится невозможным расчет модели в ускоренном режиме (**Acceleration mode**). Чтобы разорвать безынерционный контур достаточно включить в линию между датчиком напряжения и источником тока фильтр с малой постоянной времени. Величина постоянной времени должна выбираться таким образом, чтобы изменение динамических свойств модели было минимальным (обычно величина постоянной времени фильтра выбирается в пределах от 10^{-8} - 10^{-6} с). Схема модели нелинейного резистора с использованием фильтра показана на Рис. 125. Модель этой схемы в приложении называется **nelinR_2.mdl**.

Временные диаграммы работы такой модели ничем не отличаются от представленных на Рис. 123 и ВАХ – от представленной на Рис. 124.

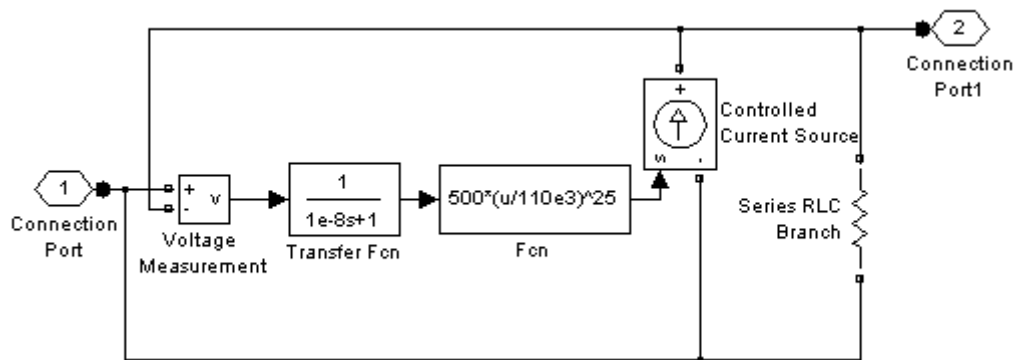


Рис. 125. Модель нелинейного резистора с использованием фильтра

3. Свойства операционного усилителя

Операционный усилитель (ОУ) представляет собой усилительное устройство с большим коэффициентом усиления K_0 ($10^3 \div 10^5$), большим входным (до сотен МОм) и малым выходным (десятки Ом) сопротивлениями и широкой полосой пропускания, начинающейся от нуля и достигающей для ряда ОУ значения сотен мегагерц. На электрических схемах ОУ обычно изображается треугольником и имеет два входа (1 и 2 на Рис. 126) и один выход (5). Под входным сопротивлением ОУ понимают входное сопротивление по одному из входов при заземленном другом. ОУ выполняются в виде интегральных микросхем, в состав которых входят несколько десятков или сотен транзисторов.

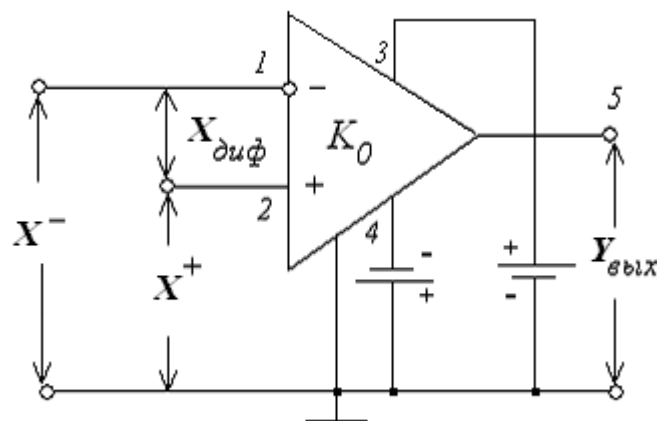


Рис. 126. Изображение операционного усилителя

Очевидно, что для того чтобы на работу усилителя не влиял источник сигнала, его выходное сопротивление должно быть существенно меньше входного сопротивления усилителя. Следствием этого является то, что входной ток усилителя очень мал. Если считать, что выходное сопротивление ис-

точника задано, то нужно стремиться к увеличению входного сопротивления усилителя. В идеале $R_{\text{вх}} \rightarrow \infty$.

Чтобы на работу усилителя не влияла нагрузка, ее сопротивление должно быть много больше выходного сопротивления усилителя, т.е. выходное сопротивление усилителя должно быть мало: $R_{\text{вых}} \rightarrow 0$.

Если подать гармоническое напряжение $x^+(t)$ на вход (+) ОУ при заземленном втором входе, то на выходе будет напряжение $K_0 x^+(t)$. Этот вход называется **неинвертирующим**. При подаче напряжения $x^-(t)$ на вход (–) на выходе получается напряжение $-K_0 x^-(t)$. Этот вход называется **инвертирующим**. Следовательно, ОУ усиливает разность сигналов $x^+(t)$ и $x^-(t)$ на его входах.

В параграфе 2.2.4. приведено определение передаточной функции или коэффициента передачи (термин, чаще используемый в русскоязычной литературе). Воспользуемся этим определением для ОУ. На Рис. 126 отмечены спектральные функции входных сигналов $X^+(i\omega)$, $X^-(i\omega)$ и выходного сигнала $Y_{\text{вых}}(i\omega)$. Можно записать:

$$Y_{\text{вых}}(i\omega) = K_0(i\omega)(X^+(i\omega) - X^-(i\omega)) = K_0(i\omega)X_{\text{диф}}(i\omega). \quad (14)$$

Поэтому такой усилитель называется дифференциальным.

В идеальном случае, если оба входа имеют одинаковый потенциал относительно «земли» (общей точки), т.е. если дифференциальное входное напряжение равно нулю, то напряжение на выходе усилителя также равно нулю, независимо от коэффициента усиления схемы.

Наличие двух входов у дифференциального усилителя дает возможность включения двух независимых цепей обратной связи, одна из которых связывает выход усилителя с его инвертирующим входом, а другая – с неинвертирующим.

Чтобы обеспечить возможность работы ОУ как с положительными, так и с отрицательными входными сигналами, используется двухполярное постоянное питающее напряжение (на Рис. 126 оно подключается к 3 и 4 выводам ОУ).

Рассмотрим некоторые параметры ОУ:

1. Дифференциальный коэффициент усиления по напряжению является комплексной величиной $\dot{K}_0 \equiv K_0 e^{i\varphi_0}$. Модуль коэффициента усиления K_0 – это отношение амплитуды сигнала на выходе к амплитуде дифференциального входного напряжения (для гармонических сигналов). Типовые значения K_0 находятся в пределах $10^3 - 10^5$.
2. Частота единичного усиления f_l . На больших частотах усиление ОУ снижается. При некоторой частоте входного сигнала усиление ОУ уменьшается до 1. Эта частота называется частотой единичного усиления. Для низкочастотных ОУ $f_l = 1$ МГц, для быстродействующих высокочастотных $f_l = 15 - 100$ МГц.

3. Входное сопротивление – сопротивление со стороны любого входа, в то время как другой вход заземлен. Входное сопротивление может для разных типов усилителей иметь величину от нескольких кОм до нескольких сотен мОм.
4. Выходное сопротивление – это сопротивление со стороны выхода в таком режиме работы, когда напряжение на входе равно нулю. Это сопротивление может быть равно 25 – 500 Ом.
5. Полоса пропускания ОУ определяется как частотный диапазон, в котором выходное напряжение уменьшается не более чем до 0,707 от своего максимального значения при неизменной амплитуде на входе. Нижняя граница полосы пропускания равна нулю (ОУ усиливают постоянные напряжения), верхняя граница полосы пропускания определяется типом ОУ и может изменяться в широких пределах.
6. Выходное напряжение сдвига – это напряжение на выходе ОУ относительно «земли» или общей точки схемы при условии, что дифференциальное входное напряжение равно нулю. Операционные усилители конструируются так, чтобы это напряжение было равно нулю. Однако практически точно выполнить это условие невозможно. Обычно, для установления выходного напряжения, равного нулю, необходимо подать на вход (между выводами «+» и «-») несколько милливольт. Некоторые ОУ имеют выводы для подключения потенциометра, регулирующего напряжение на выходе при отсутствии входного сигнала.
7. Входное напряжение смещения – напряжение, которое нужно приложить между входами усилителя для получения нуля на выходе. Эта величина чаще всего лежит в диапазоне 0 – 10 мВ.
8. Максимальное выходное напряжение – амплитуда выходного напряжения относительно нуля при работе усилителя на линейном участке. У разных типов ОУ максимальное выходное напряжение лежит в диапазоне $\pm(3 - 15)$ В.

Для расчетов схем с ОУ широко используют понятие об *идеальном операционном усилителе*, подразумевая под этим, что у ОУ:

- коэффициенты усиления по току и напряжению и входное сопротивление каждого из входов ОУ бесконечно велики;
- усилитель полностью симметричен, не имеет дрейфа нуля и усиливает одинаково колебания любых частот. Дрейфом нуля называется изменение выходного напряжения, не связанные со входным сигналом и обусловленные внутренними процессами в усилителе. Главная причина дрейфа – изменение температуры.

При дальнейшем изложении мы будем рассматривать ОУ как идеальный.

4. Моделирование операционного усилителя в MATLAB

В библиотеке **SimPowerSystems** нет специального блока для операционного усилителя. В этом случае, как уже обсуждалось выше (п. 2.10.5.), пользователь имеет возможность создать свой собственный блок, как с помощью уже имеющихся в библиотеке блоков, реализуя возможности **Simulink** по созданию подсистем, так и на основе блоков основной библиотеки **Simulink** и управляемых источников тока или напряжения. Рассмотрим, какой моделью можно описать ОУ. Исходя из частотных свойств, ОУ является фильтром низких частот с широкой полосой пропускания. Поэтому моделировать частотные свойства ОУ будем с помощью функции передачи фильтра низких частот и коэффициента усиления:

$$\dot{K} = \frac{K_0}{1 + i\omega\tau}, \quad (15)$$

где K_0 – коэффициент передачи ОУ на постоянном токе (коэффициент усиления). Модуль передаточной функции (15) будет:

$$|\dot{K}| = \frac{K_0}{\sqrt{1 + \omega^2\tau^2}} = \frac{K_0}{\sqrt{1 + 4\pi^2 f^2\tau^2}}$$

Учитывая, что на частоте единичного усиления f_1 модуль коэффициента передачи равен единице, находим постоянную времени цепочки τ :

$$\tau = \sqrt{\frac{K_0^2 - 1}{4\pi^2 f_1^2}}. \quad (16)$$

Обычно коэффициент усиления K_0 очень большой, поэтому единицей в числителе выражения (16) можно пренебречь. Тогда, коэффициент передачи (15) можно переписать как:

$$\dot{K} = \frac{K_0}{1 + i\omega \frac{K_0}{2\pi f_1}} \quad (17)$$

Таким образом, функция передачи ОУ будет описываться двумя параметрами K_0 и f_1 . Для типичных значений $K_0 = 10^5$ и $f_1 = 10^8/2\pi$ Гц зависимость модуля (**Magnitude**) и фазы (**Phase**) комплексной функции передачи от частоты (**Frequency**) будет выглядеть следующим образом (Рис. 127):

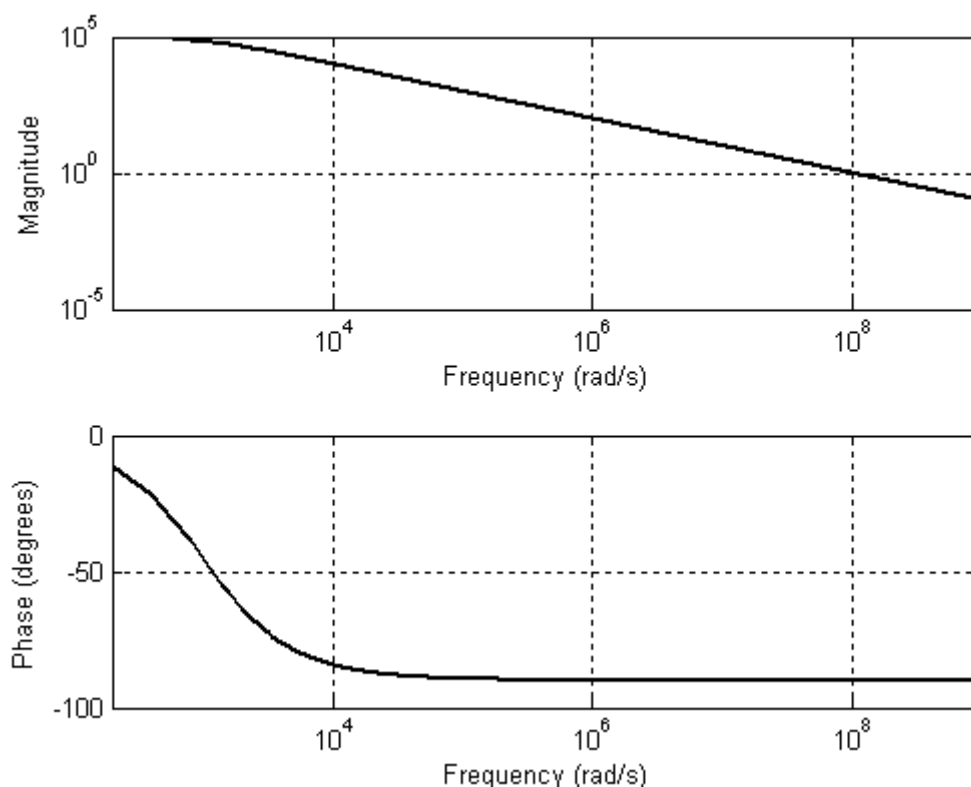


Рис. 127. АЧХ и ФЧХ модели ОУ

Для наглядности графики приведены в логарифмическом масштабе. Видно, что при циклической частоте $\omega_l = 10^8$ рад/с модуль функции передачи равен 1.

Рассмотрим модель ОУ в **SimPowerSystems** (Рис. 128). В приложении эта модель называется **OU.mdl**. Взаимодействие усилителя с электрической цепью происходит через клеммы 1, 3, обозначенные соответственно «+» и «-» (вход ОУ), и клемму 2, обозначенную «out» (выход ОУ). Соединение с **Simulink**-моделью осуществляется с помощью двух блоков измерителей напряжения **Voltage Measurement** (свой для каждого входа) и блока управляемого источника напряжения **Controlled Voltage Source**. Три сопротивления на схеме моделируют входные сопротивления ОУ (по каждому входу) и выходное сопротивление ОУ. Основу модели составляет схема из блоков **Simulink**.

Коэффициент передачи ОУ на постоянном токе задает усилитель **Gain** (коэффициент усиления в окне параметров блока задается переменной $K0$), на вход которого поступает разность входных напряжений усилителя с блока **Add**. Блок передаточной функции **Transfer Fcn** задает частотные свойства операционного усилителя (вектор числителя блока [1], а знаменателя $[K0/2/\pi/f_l \ 1]$), а блок насыщения **Saturation** обеспечивает ограничение выходного напряжения на уровне напряжений положительной и отрицательной полярностей (величины насыщения в свойствах блока обозначены переменными lim и $-lim$), имитируя нелинейность характеристики реального усилителя вызванную, конечной величиной напряжений питания.

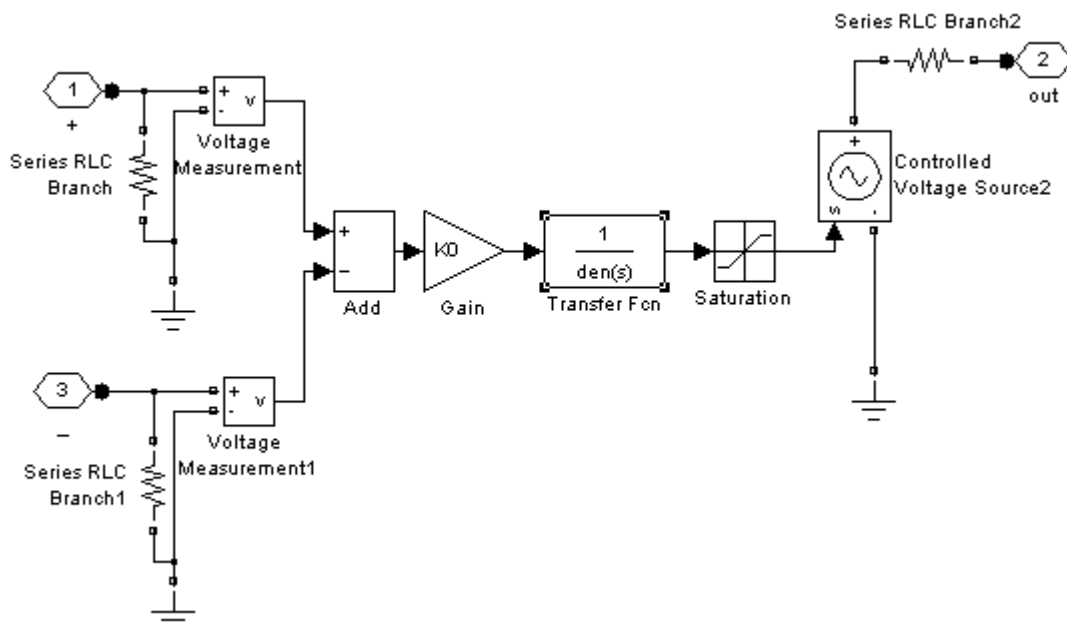


Рис. 128. Модель ОУ

Оформим модель ОУ как отдельный блок, чтобы его можно было в дальнейшем использовать для моделирования различных электрических схем. Для этого создадим маскированную подсистему, отражающую общепринятое условное обозначение ОУ в схеме и параметры, характеризующие модель ОУ. Окно с вкладкой **Icon**, описывающей внешний вид блока показано на Рис. 129. Здесь в основном окне введена команда **MATLAB plot**, соединяющая прямыми отрезками точки, координаты которых указаны в параметрах команды. В данном случае, это вектор координат $X [10 \ 10 \ 40 \ 10]$ и соответствующий вектор координат $Y [10 \ 20 \ 15 \ 10]$. Эта команда рисует треугольник. Чтобы не было видно квадрата границы самой иконки подсистемы, в окне **Frame** была выбрана опция **Invisible** (невидимый). Для того чтобы были видны порты подсистемы и их обозначения на иконке, в окне **Transparency** выбирается опция **Transparent** (прозрачный).

Окно параметров **Parameters** показано на Рис. 130. Здесь введены основные параметры, описывающие модель ОУ:

- **gain** – коэффициент передачи на постоянном токе;
- **unit frequency** – частота единичного усиления;
- **saturation** – пределы насыщения (усилитель симметричный).

После создания маски блок ОУ будет выглядеть, как показано на Рис. 131. На этом же рисунке приведено окно параметров этого блока, оно будет открываться при двойном щелчке по иконке блока вместо схемы подсистемы.

В приближении идеального ОУ можно упростить схему Рис. 128.

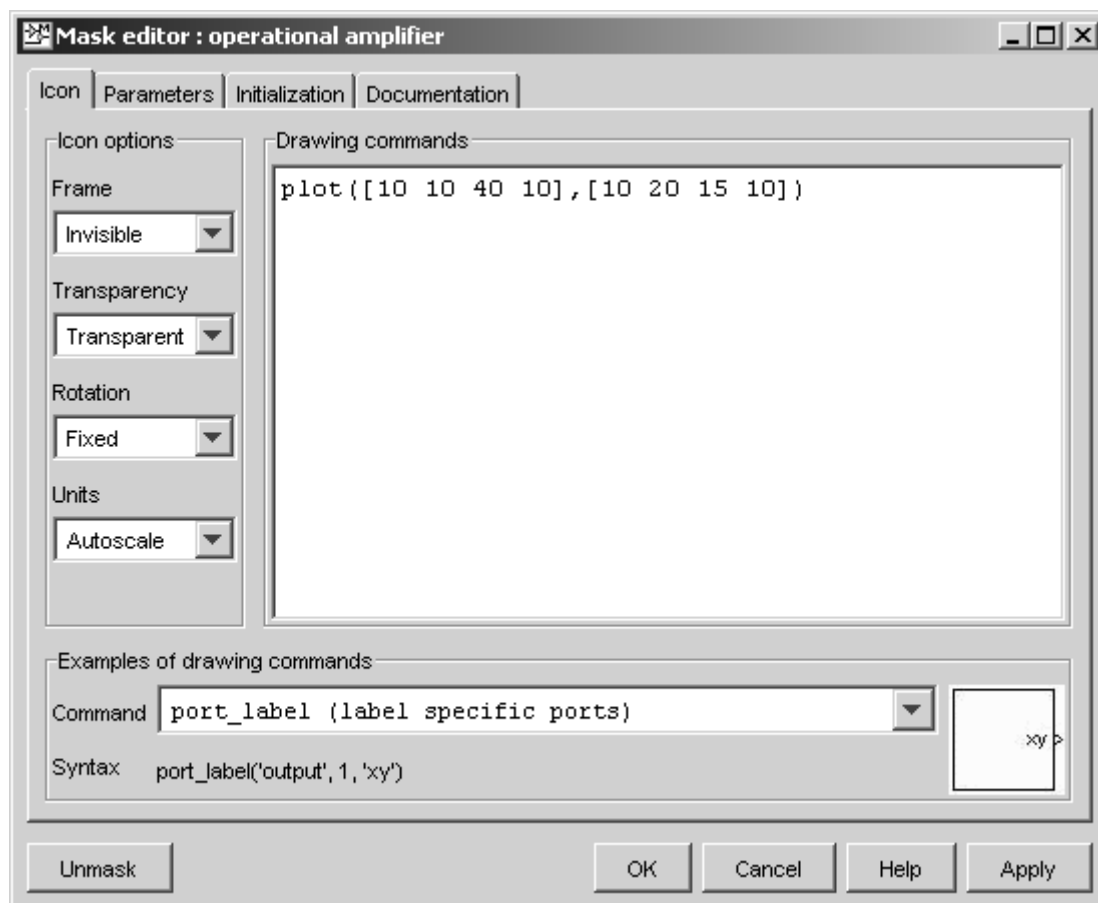


Рис. 129. Окно с вкладкой Icon редактора маски ОУ

Можно отказаться от резисторов считая, что входное сопротивление идеального ОУ бесконечно велико, а выходное – бесконечно мало. Также, если не учитывается частотная зависимость, можно исключить блок передаточной функции.

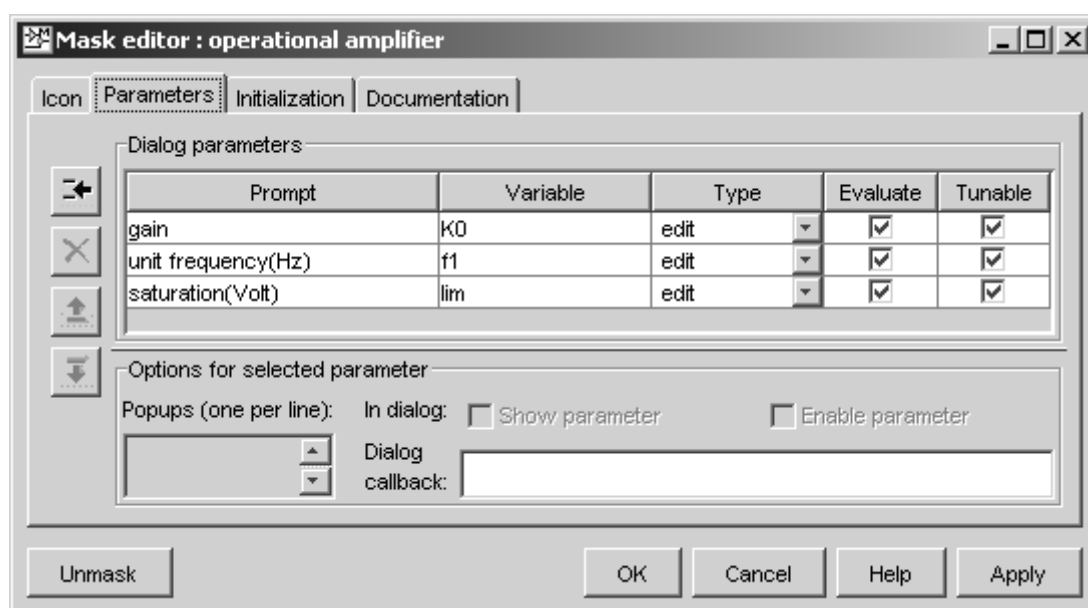


Рис. 130. Окно с вкладкой Parameters редактора маски ОУ

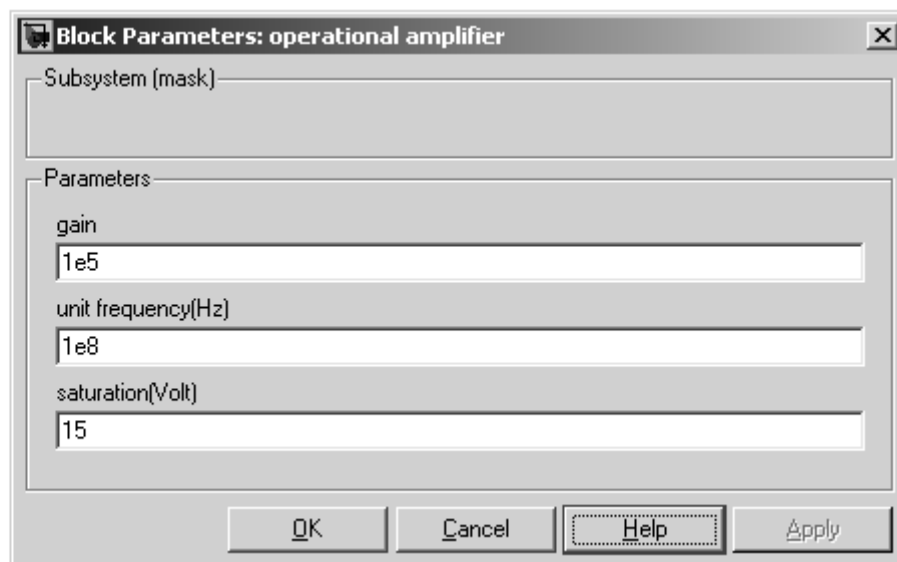
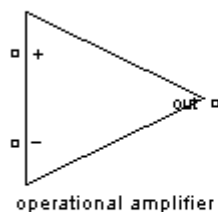


Рис. 131. Иконка ОУ и окно параметров ОУ

5. Применение операционного усилителя в различных схемах

Мы рассмотрим три области применения схем с ОУ:

- фильтрация сигналов;
- усиление сигналов;
- генерация сигналов.

5.1. Активные фильтры

При работе с электрическими сигналами часто требуется выделить из них какую-либо одну частоту или полосу частот (например, разделить шумовой и полезный сигналы). Для подобного разделения используются электрические фильтры.

Как хорошо известно, для этой цели можно использовать пассивные фильтры, которые выполняются на RC - или RLC -цепях без применения активных усилительных элементов. Используя различные комбинации включения звеньев первого и второго порядков, состоящих из RLC -элементов, можно реализовать практически любую амплитудно-частотную и фазочастотную характеристики, удовлетворяющие условиям физической осуществимости.

Однако пассивные фильтры имеют несколько недостатков. При простом каскадном соединении RLC -звеньев коэффициент передачи цепи не будет равен произведению коэффициентов передачи отдельных каскадов, т.к. входное сопротивление последующего звена будет служить нагрузкой для предыдущего. Если же желательно получить их произведение, то между ними необходимо включать разделительные каскады с большими входными и

малыми выходными сопротивлениями и с независимыми от частоты коэффициентами передачи (например, операционные усилители). Также, в состав цепей, имеющих резкие экстремумы АЧХ, входят катушки индуктивности, технология изготовления которых не отвечает современным требованиям к микросхемам: катушки индуктивности имеют большие габариты и массу, в них рассеивается большая мощность, в катушках с сердечником проявляются нелинейные эффекты и т.д. Эти трудности также могут быть преодолены с использованием операционных усилителей, охваченных одной или несколькими цепями обратной связи, содержащими только RC -элементы.

Активный фильтр представляет собой четырехполюсник, содержащий пассивные RC -цепи и активные элементы: транзисторы, электронные лампы или операционные усилители. Активные фильтры обычно не содержат катушек индуктивности. В отличие от пассивных RC -фильтров, активные фильтры обеспечивают более качественное разделение полос пропускания и затухания. В них сравнительно просто можно регулировать неравномерности частотной характеристики в области пропускания и затухания.

В данном методическом пособии будет рассказано о некоторых типовых схемах фильтров, создаваемых на основе операционных усилителей, а также будут приведены методы их расчета и компьютерного моделирования.

5.1.1. Основные сведения о фильтрах

Основной характеристикой фильтра является его АЧХ, т.е. зависимость модуля коэффициента передачи K от частоты сигнала. Для идеальных фильтров существуют частоты среза, разделяющие области пропускания и задерживания, а затухание фильтра скачкообразно изменяется при переходе от полосы пропускания к полосе задерживания. В реальных фильтрах полосы пропускания и задерживания разделяются переходной зоной, в которой коэффициент передачи изменяется непрерывно от максимального до минимального значения. Вместо частот среза появляются две граничные частоты: частота среза полосы пропускания (ω_1) и частота среза полосы задерживания (ω_2). Чем меньше отличаются эти частоты, тем выше качество фильтра.

Электрические фильтры, которые описываются линейными дифференциальными уравнениями с постоянными коэффициентами, описываются передаточными функциями вида:

$$\dot{K}(s) = \frac{a_m s^m + a_{m-1} s^{m-1} + \dots + a_0}{b_n s^n + b_{n-1} s^{n-1} + \dots + b_0}, \text{ причем } m < n.$$

Здесь $s = \sigma + i\omega$. Передаточная функция $K(s)$ полностью задается совокупностью корней числителя (нулей) и знаменателя (полюсов). Число полюсов передаточной функции определяет порядок фильтра. Таким образом, можно описать и активные фильтры с идеальным операционным усилителем, который мы моделируем с помощью коэффициента передачи фильтра нижних частот.

Активные фильтры можно разделить на группы по различным признакам: назначению, полосе пропускаемых частот, типу усилительных элементов, виду обратных связей и др. По полосе пропускаемых частот фильтры делятся на четыре основные группы: нижних частот, верхних частот, полосовые и заграждающие. Фильтры нижних частот пропускают сигналы от постоянного напряжения до некоторой предельной частоты, называемой частотой среза фильтра. Фильтры верхних частот – начиная с частоты среза и выше. Полосовые фильтры пропускают сигналы в некоторой полосе частот от f_1 до f_2 , а заграждающие фильтры имеют характеристику, противоположную полосовым, и пропускают сигналы с частотой ниже f_1 и выше f_2 . Как полосовые, так и заграждающие фильтры могут иметь гребенчатую частотную характеристику, в которой будет несколько полос пропускания и затухания.

По назначению фильтры делятся на сглаживающие фильтры источников питания, заграждающие фильтры помех, фильтры для селективных усилителей низкой или высокой частоты и др.

По типу усилительных элементов можно выделить транзисторные фильтры, фильтры на усилителях с ограниченным усилением, на операционных усилителях, на повторителях напряжения и др. Все рассмотренные фильтры могут иметь одну цепь обратной связи или несколько. В связи с этим различают фильтры с одноконтурной и многоконтурной обратной связью. Кроме этого, различают фильтры по числу полюсов на частотной характеристике – фильтры первого порядка, второго и более высоких порядков. Фильтры высоких порядков имеют более крутые границы полос пропускания и затухания и более плоскую характеристику в области полосы пропускания, что естественно улучшает качество фильтра. К таким фильтрам относятся фильтры Чебышева, Баттерворта, Бесселя и др.

5.1.2. Понятие обратной связи

Обратные связи позволяют в широких пределах управлять входными и выходными характеристиками четырехполюсников, а также их частотными свойствами и величиной усиления. Обратная связь, увеличивающая модуль коэффициента передачи по сравнению с собственным коэффициентом передачи ОУ K_0 , называется положительной, а уменьшающая его – отрицательной.

Рассмотрим принцип введения отрицательной обратной связи (Рис. 132) для идеального ОУ. На схеме $X(s)$ и $Y(s)$ обозначены спектральные плотности входного и выходного сигналов соответственно. На вход ОУ подается разность входного сигнала и выходного сигнала, прошедшего через цепь обратной связи с коэффициентом передачи $\beta(s)$. Спектральная плотность на выходе цепи обратной связи равна произведению коэффициента передачи этой цепи $\beta(s)$ на спектральную плотность сигнала, который подается на вход цепи обратной связи (т.е. выходного сигнала $Y(s)$) $\beta(s)Y(s)$.

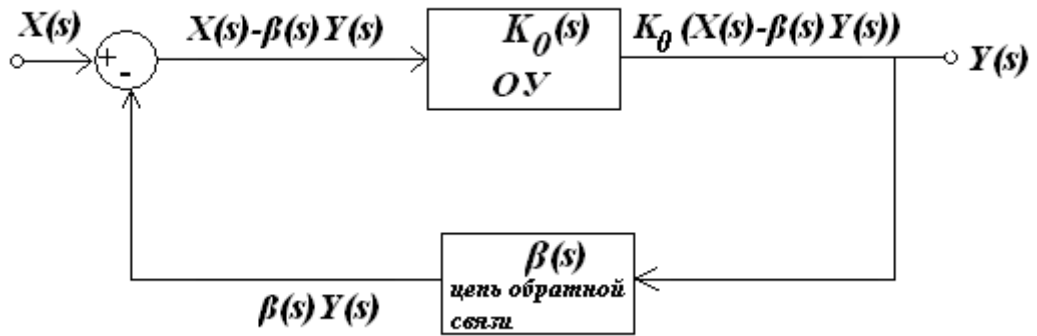


Рис. 132. Схема, поясняющая принцип действия отрицательной обратной связи

Аналогично, спектральная плотность сигнала на выходе ОУ равна:

$$Y(s) = K_0(X(s) - \beta(s)Y(s)) \quad (18)$$

Тот факт, что выходное напряжение воздействует на входное напряжение, причем так, что это влияние направлено в сторону, противоположную изменениям входной величины, и есть проявление отрицательной обратной связи. В выражении (18) раскрываем скобки и переносим слагаемые с $Y(s)$ в левую часть:

$$Y(s)(1 + K_0\beta(s)) = K_0X(s) \quad (19)$$

Из этого выражения легко получить коэффициент передачи цепи с обратной связью $K(s)$:

$$K(s) = \frac{Y(s)}{X(s)} = \frac{K_0}{1 + K_0\beta(s)}. \quad (20)$$

Выражение (20) позволяет сделать несколько важных выводов. Во-первых, от свойств цепи обратной связи существенно зависит величина коэффициента передачи усилителя. В случае положительной обратной связи уравнение (20) можно записать как $K(s) = \frac{K_0}{1 - K_0\beta(s)}$. При этом должно выполняться условие $|K_0\beta(s)| < 1$, иначе возможно самовозбуждение.

Необходимо отметить, что одна и та же схема включения обратной связи на одних частотах может увеличивать общий коэффициент передачи, а на других – ослаблять. Входной сигнал может быть представлен в виде линейной комбинации гармонических функций (ряд Фурье). Каждая гармоника при прохождении через усилитель и цепь обратной связи будет иметь суммарный сдвиг фаз $\varphi_\Sigma = \varphi_0 + \varphi_\beta$, где $\varphi_\beta = \arg\beta$. Для каждой гармоники K_0 и $|\beta|$ фиксированы (т.к. частота гармоники постоянна). В зависимости от величины суммарного сдвига фазы гармоника может либо усиливаться, либо подавляться.

Максимальное увеличение амплитуды гармоники происходит при $\varphi_{\Sigma} = 2\pi k$ (положительная обратная связь), максимальное же уменьшение будет при $\varphi_{\Sigma} = \pi + 2\pi k$ (отрицательная обратная связь).

Во-вторых, обратная связь может существенным образом влиять на частотные свойства активного четырехполюсника. В том случае, когда $|K_0\beta(s)| \gg 1$, зависимость $K(s)$ практически полностью определяется частотными свойствами цепи обратной связи. При $K_0 \gg 1$ (что выполняется для ОУ) коэффициент усиления охваченного обратной связью (из формулы (20)) усилителя составит

$$K(s) \approx \frac{1}{\beta(s)}.$$

Таким образом, из этого соотношения следует, что коэффициент усиления усилителя с обратной связью определяется только обратной связью и не зависит от параметров самого усилителя.

В качестве примера рассмотрим наиболее часто встречающуюся схему включения ОУ (Рис. 133).

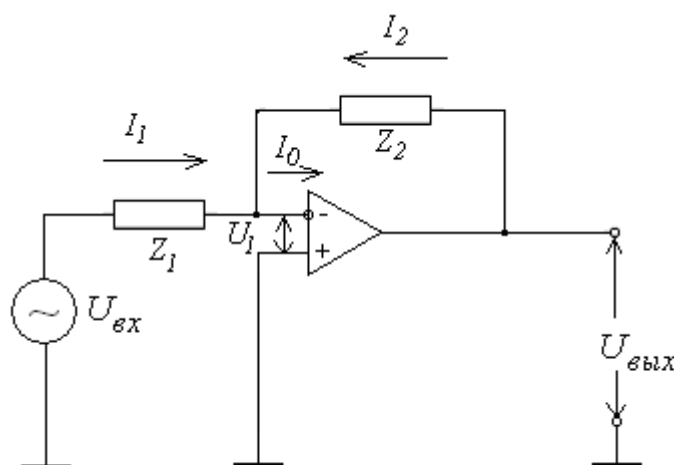


Рис. 133. Пример включения ОУ

Будем считать, что входной сигнал – гармонический $\dot{U}_{\text{вх}}(t) = \dot{U}_0 e^{i\omega t}$, где \dot{U}_0 – комплексная амплитуда источника напряжения. Тогда выходной сигнал также будет гармоническим $\dot{U}_{\text{вых}}(t) = \dot{U}_2 e^{i\omega t}$. Поскольку дифференциальный коэффициент усиления идеального ОУ $K_0 = \infty$, а выходное напряжение ограничено, то из (14) можно считать, что $\dot{U}_1 = 0$, и токи определяются из законов Кирхгофа для комплексных амплитуд:

$$\dot{I}_1 = \frac{\dot{U}_0 - \dot{U}_1}{Z_1} = \frac{\dot{U}_0}{Z_1}, \quad (21)$$

$$\dot{I}_2 = \frac{\dot{U}_2 - \dot{U}_1}{Z_2} = \frac{\dot{U}_2}{Z_2}. \quad (22)$$

Ввиду того, что $\dot{U}_1 = 0$, входной ток ОУ $\dot{I}_0 = \frac{\dot{U}_1}{R_{\text{вх}}}$ также равен нулю и, учитывая закон Кирхгофа для комплексных амплитуд токов $\dot{I}_1 + \dot{I}_2 = \dot{I}_0$, получим $\dot{I}_1 = -\dot{I}_2$.

Выражая токи с помощью (21) и (22), получаем коэффициент усиления схемы:

$$\dot{K} = \frac{\dot{U}_{\text{вых}}}{\dot{U}_{\text{вх}}} = -\frac{Z_2}{Z_1}. \quad (23)$$

Знак минус означает, что входной и выходной сигналы находятся в противофазе. Взяв в качестве комплексных сопротивлений Z_1 и Z_2 сопротивления R_1 и R_2 , получим широкополосный инвертирующий усилитель. Если положить $Z_1 = R$, а $Z_2 = 1/i\omega C$, получим интегратор, а при $Z_1 = 1/i\omega C$ и $Z_2 = R$ – дифференциатор.

Прежде чем перейти к рассмотрению конкретных схем включения ОУ, уточним, в каких случаях при расчете внешних элементов можно пользоваться понятием идеальности ОУ. Очевидно, что током во входной цепи усилителя (Рис. 133) можно пренебречь, если входное сопротивление ОУ много больше сопротивления параллельно соединенных Z_1 и Z_2 ; коэффициент усиления по напряжению можно считать бесконечным, если он много больше требуемого коэффициента усиления, определяемого цепью обратной связи; выходное сопротивление можно считать бесконечно малым, если оно много меньше сопротивления нагрузки усилителя. Итак, при всех расчетах, которые проводятся в предположении идеальности ОУ, следует выбирать такие сопротивления, чтобы выполнялись следующие условия:

$$K_0 \gg \left| \frac{Z_2}{Z_1} \right|; |Z_{\text{вх}}| \gg \left| \frac{Z_1 Z_2}{Z_1 + Z_2} \right|; |Z_{\text{вых}}| \ll |Z_H|$$

5.1.3. Основные типы активных фильтров: схема первого типа

После сделанных предварительных замечаний займемся рассмотрением основных типов активных фильтров на ОУ. В дальнейшем мы будем рассматривать схемы, у которых частотнозависимой является только обратная связь $\beta(s)$. По этому признаку можно выделить два типа схем. В схеме первого типа частотно зависящая обратная связь подается на неинвертирующий вход ОУ, а в схеме второго типа частотнозависимая обратная связь подается на инвертирующий вход ОУ, а частотнонезависимая – на неинвертирующий. Блок-схема активного фильтра первого типа изображена на Рис. 134.

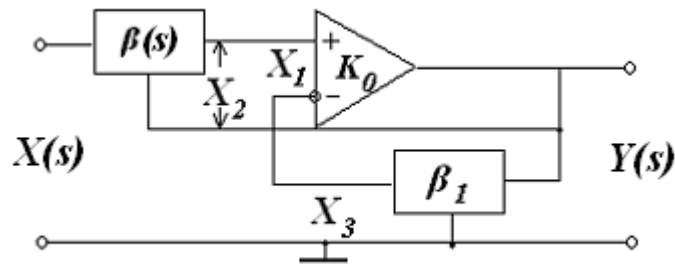


Рис. 134. Блок-схема активного фильтра первого типа

Используя обозначения рисунка, получим коэффициент передачи этой схемы на основании следующих соотношений:

$$\begin{aligned}
 Y(s) &= K_0 X_1(s) = K_0 [X_2(s) + Y(s) - X_3(s)] = \\
 &= K_0 [(X(s) - Y(s))\beta(s) + Y(s) - \beta_1 Y(s)] = \\
 &= K_0 \beta(s) X(s) + K_0 (1 - \beta(s) - \beta_1) Y(s)
 \end{aligned} \tag{24}$$

Отсюда

$$K(s) = \frac{Y(s)}{X(s)} = \frac{\beta(s)K_0}{1 + \beta_1 K_0 + (\beta(s) - 1)K_0} \tag{25}$$

Преобразуем формулу (25) следующим образом:

$$K(s) = \frac{K_0}{(1 + \beta_1 K_0)} \cdot \frac{\beta(s)}{1 + (\beta(s) - 1) \frac{K_0}{1 + \beta_1 K_0}} = \frac{\beta(s)K_p}{1 + (\beta(s) - 1)K_p} \tag{26}$$

где $K_p = \frac{K_0}{1 + \beta_1 K_0}$. При $\beta_1 K_0 \gg 1$ имеем $K_p \approx 1/\beta_1$.

Рассмотрим конкретный пример активного фильтра, построенного по схеме первого типа (Рис. 135). $X^+(s)$ – комплексная амплитуда напряжения на неинвертирующем входе ОУ (относительно «земли»).

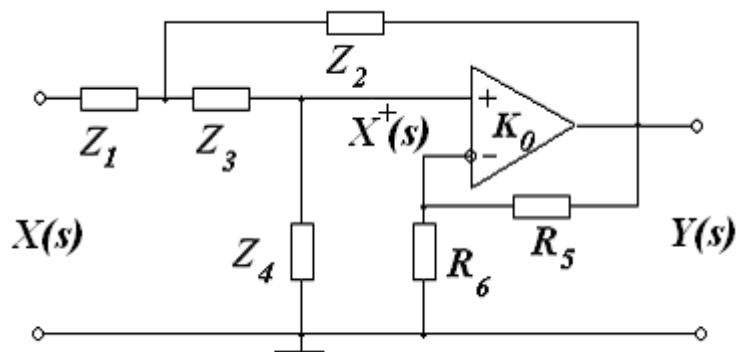


Рис. 135. Пример активного фильтра первого типа

Данную схему можно представить в виде схемы с двумя генераторами: генератором входного напряжения $x(t)$ и управляемым (зависимым) генератором выходного напряжения $y(t)$ (Рис. 136). Входное сопротивление ОУ бесконечно велико, поэтому через него ток не течет (входы и выход ОУ можно считать разомкнутыми). На эквивалентной схеме не показана ветвь цепи, состоящая из сопротивлений R_5 и R_6 . Соотношение этих сопротивлений влияет на величину напряжения на инвертирующем входе ОУ, и будет учитываться при выводе.

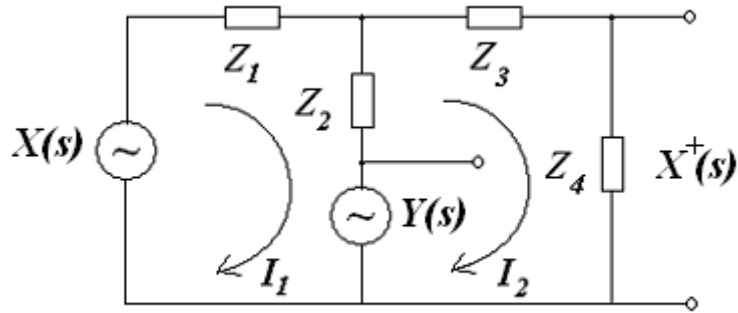


Рис. 136. Эквивалентная схема

Связь между L -изображениями токов и напряжений в схеме на Рис. 136 описывается следующей системой уравнений:

$$\begin{aligned} X(s) - Y(s) &= (Z_1 + Z_2)I_1(s) - Z_2I_2(s), \\ Y(s) &= -Z_2I_1(s) + (Z_2 + Z_3 + Z_4)I_2(s) \end{aligned} \quad (27)$$

Найдем ток $I_2(s)$:

$$I_2(s) = \frac{Z_1Y(s) + Z_2X(s)}{\Delta}, \quad (28)$$

где $\Delta = (Z_1 + Z_2)(Z_2 + Z_3 + Z_4) - Z_2^2 = (Z_1 + Z_2)(Z_3 + Z_4) + Z_1Z_2$ — определитель системы уравнений. Зная $I_2(s)$, можно определить L -преобразование сигнала $x^+(t)$:

$$X^+(s) = Z_4I_2(s). \quad (29)$$

Используя равенство $y(t) = K_0[x^+(t) - x^-(t)]$ и учитывая, что в нашем случае L -изображение напряжения на инвертирующем входе ОУ равно

$$X^-(s) = Y(s) \frac{R_6}{R_5 + R_6}, \quad (30)$$

имеем

$$Y(s) = K_0[X^+(s) - X^-(s)] = K_0 \left[X^+(s) - Y(s) \frac{R_6}{R_5 + R_6} \right]. \quad (31)$$

Следовательно,

$$Y(s) = \frac{K_0}{1 + \beta K_0} X^+(s) = K_p X^+(s), \quad (32)$$

где

$$\beta = \frac{R_6}{R_5 + R_6} \text{ и } K_p = \frac{K_0}{1 + \beta K_0}.$$

Используя формулы (28) и (29), получаем искомый коэффициент передачи активного фильтра, построенного по схеме первого типа:

$$K(s) = \frac{Y(s)}{X(s)} = K_p \frac{Z_2 Z_4}{\Delta - Z_1 Z_4 K_p} \quad (33)$$

Покажем теперь, что соответствующим выбором элементов Z_1, Z_2, Z_3, Z_4 в этой схеме можно реализовать как фильтр низких, так и фильтр высоких частот второго порядка без применения индуктивностей, а также режекторный фильтр.

5.1.4. Активный RC-фильтр низких частот второго порядка

Выберем параметры Z_1, Z_2, Z_3, Z_4 следующим образом:
 $Z_1 = R_1, Z_2 = 1/sC_2, Z_3 = R_3, Z_4 = 1/sC_4$. Схема в этом случае примет вид, изображенный на Рис. 137.

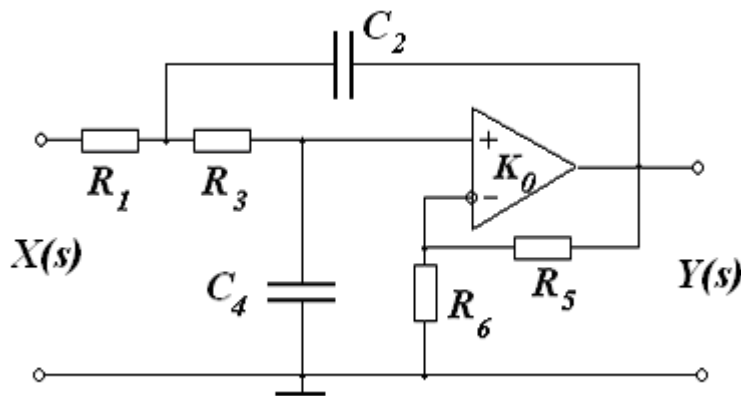


Рис. 137. Схема активного RC-фильтра низких частот

Прямой подстановкой значений параметров Z_k в выражение (33) получим в стандартных обозначениях:

$$K(s) = K_p \frac{\omega_0^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}, \quad (34)$$

$$\text{где } \omega_0^2 = \frac{1}{R_1 R_3 C_2 C_4} \text{ и } Q = \frac{\sqrt{\frac{R_3 C_2}{R_1 C_4}}}{1 + \frac{R_3}{R_1} + \frac{C_2}{C_4}(1 - K_p)}. \quad (35)$$

При $K_0 \gg 1$

$$K_p = \frac{K_0}{1 + \beta K_0} = \frac{K_0}{1 + \frac{R_6}{R_5 + R_6} K_0} \approx 1 + \frac{R_5}{R_6}, \quad (36)$$

тогда формула для Q в(35) будет:

$$Q = \frac{\sqrt{\frac{R_3 C_2}{R_1 C_4}}}{1 + \frac{R_3}{R_1} - \frac{R_5 C_2}{R_6 C_4}}. \quad (37)$$

Определим корни знаменателя выражения (34):

$$s_{1,2} = -\frac{\omega_0}{2Q} \left(1 \pm \sqrt{1 - 4Q^2} \right). \quad (38)$$

Коэффициент передачи $K(s)$ в (34) оказывается таким же, как и у пассивного фильтра второго порядка, содержащего все три элемента (R, L, C) (Рис. 138), для которого:

$$\omega_0^2 = \frac{1}{LC}, \quad Q = \frac{\omega_0 L}{R}, \quad K_p = 1.$$

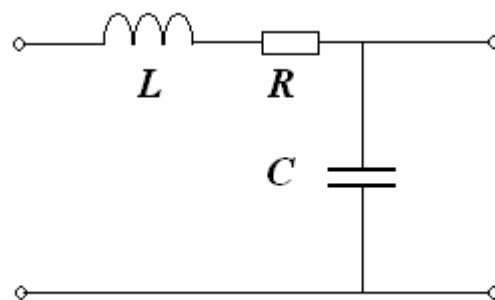


Рис. 138. Пассивный RLC -фильтр

Исследовать АЧХ и ФЧХ рассматриваемого активного фильтра можно, например, в **Simulink**, используя блок передаточной функции (или непосредственно с помощью команды **MATLAB freqs**). Для параметров электриче-

ской схемы $K_p = 1$, $\omega_0 = 200$ рад/с и $Q = 10$ **Simulink**-модель с блоком передаточной функции будет выглядеть, как показано на Рис. 139. В приложении эта модель называется **filter1_low_transferfnc.mdl**.

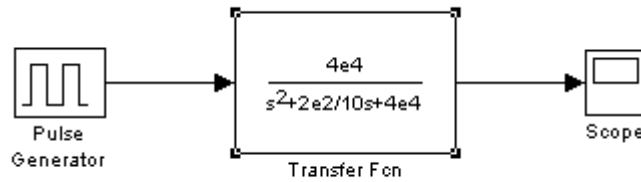


Рис. 139. Simulink-модель активного фильтра низких частот

АЧХ и ФЧХ можно получить с помощью **LTI-viewer**. Но в данном случае проще использовать команду **MATLAB freqs**. Ниже приведен листинг для получения графиков АЧХ и ФЧХ (**filter1_low_transferfnc_mat.m**).

```
w0=2e2; %собственная частота
Q=10; %добротность
w=0:1:400; %диапазон частот
%вектор числителя передаточной функции:
b=[w0^2];
%вектор знаменателя передаточной функции:
a=[1 w0/Q w0^2];
freqs(b,a,w); %расчет и построение АЧХ и ФЧХ
```

Зависимость частотных характеристик от величины параметра Q иллюстрируется Рис. 140. Из рисунка видно, что с ростом Q проявляется резонансный характер амплитудно-частотной характеристики.

Рассчитаем теперь переходную характеристику $h(t)$ этой схемы – ее реакцию на функцию единичного включения $1_+(t)$:

$$1_+(t) = \begin{cases} 0, & t < 0, \\ 1, & t \geq 0. \end{cases} \quad (39)$$

Учитывая, что L -изображение функции $1_+(t)$ равно $1/s$, запишем:

$$h(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{\dot{K}(s)}{s} e^{st} ds = \frac{\omega_0^2 K_p}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{e^{st}}{s(s-s_1)(s-s_2)} ds, \quad (40)$$

где s_1 и s_2 определяются выражением (38). Разложим дробь, входящую в подынтегральное выражение (40), на простейшие:

$$\frac{1}{s(s-s_1)(s-s_2)} = \frac{1}{ss_1s_2} + \frac{1}{s_1(s_1-s_2)} \frac{1}{(s-s_1)} + \frac{1}{s_2(s_2-s_1)} \frac{1}{(s-s_2)}.$$

Тогда
$$h(t) = \frac{1}{2\pi i} \left[\frac{\omega_0^2 K_p}{s_1 s_2} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{e^{st}}{s} ds + \frac{\omega_0^2 K_p}{s_1 (s_1 - s_2)} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{e^{st}}{(s - s_1)} ds + \frac{\omega_0^2 K_p}{s_2 (s_2 - s_1)} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{e^{st}}{(s - s_2)} ds \right]. \quad (41)$$

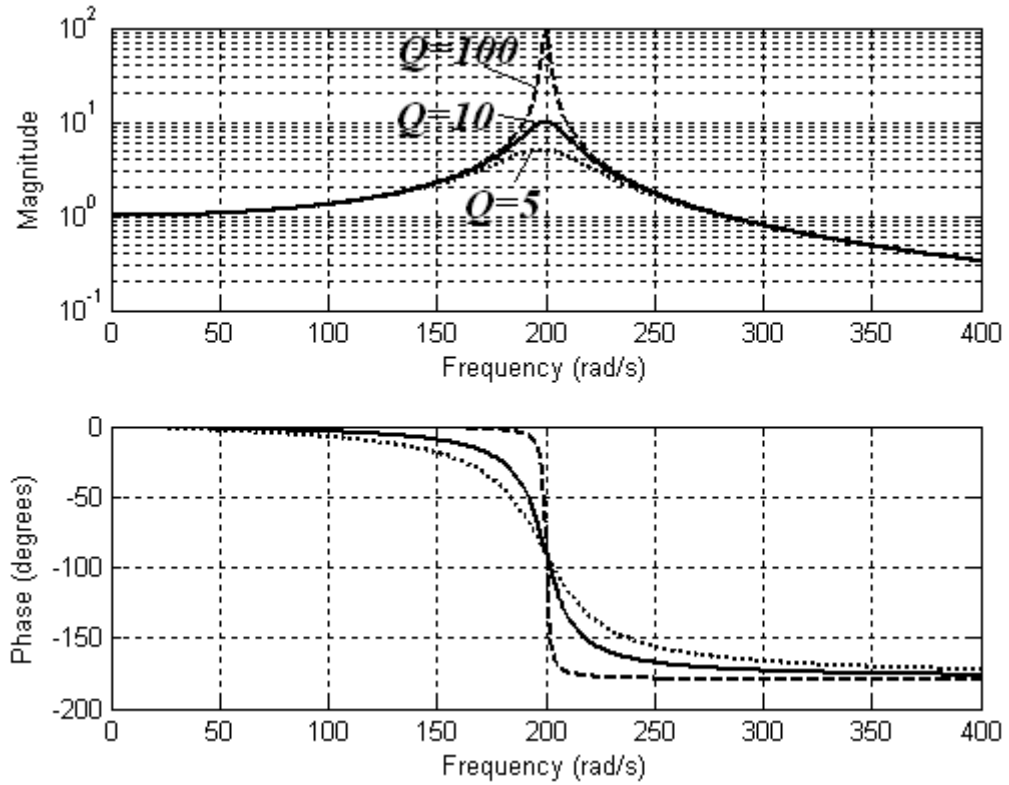


Рис. 140. АЧХ и ФЧХ активного фильтра низких частот для разных Q

Для вычисления интегралов, входящих в формулу (41), найдем L -изображение функции $e^{\alpha t}$:

$$\int_0^{\infty} e^{\alpha t} e^{-st} dt = \frac{1}{\alpha - s} e^{(\alpha-s)t} \Big|_0^{\infty} = \frac{1}{s - \alpha}$$

Следовательно,

$$\frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{e^{st}}{s - \alpha} ds = e^{\alpha t}. \quad (42)$$

Поэтому переходная характеристика рассматриваемого фильтра имеет вид:

$$\begin{aligned} h(t) &= \frac{\omega_0^2 K_p}{s_1 s_2} + \frac{\omega_0^2 K_p}{s_1 (s_1 - s_2)} \exp(s_1 t) + \frac{\omega_0^2 K_p}{s_2 (s_2 - s_1)} \exp(s_2 t) = \\ &= K_p + \frac{\omega_0^2 K_p}{(s_1 - s_2)} \left(\frac{\exp(s_1 t)}{s_1} - \frac{\exp(s_2 t)}{s_2} \right) \end{aligned} \quad (43)$$

Вид зависимости $h(t)$ (Рис. 141) определяется значением параметра Q . При $Q > 0,5$ переходной процесс имеет осциллирующий характер. «Размах» колебаний увеличивается, а их затухание уменьшается с ростом добротности.

При $0 < Q < 0,5$ выходной сигнал нарастает аperiodически. Установившееся предельное значение выходного сигнала не зависит от Q .

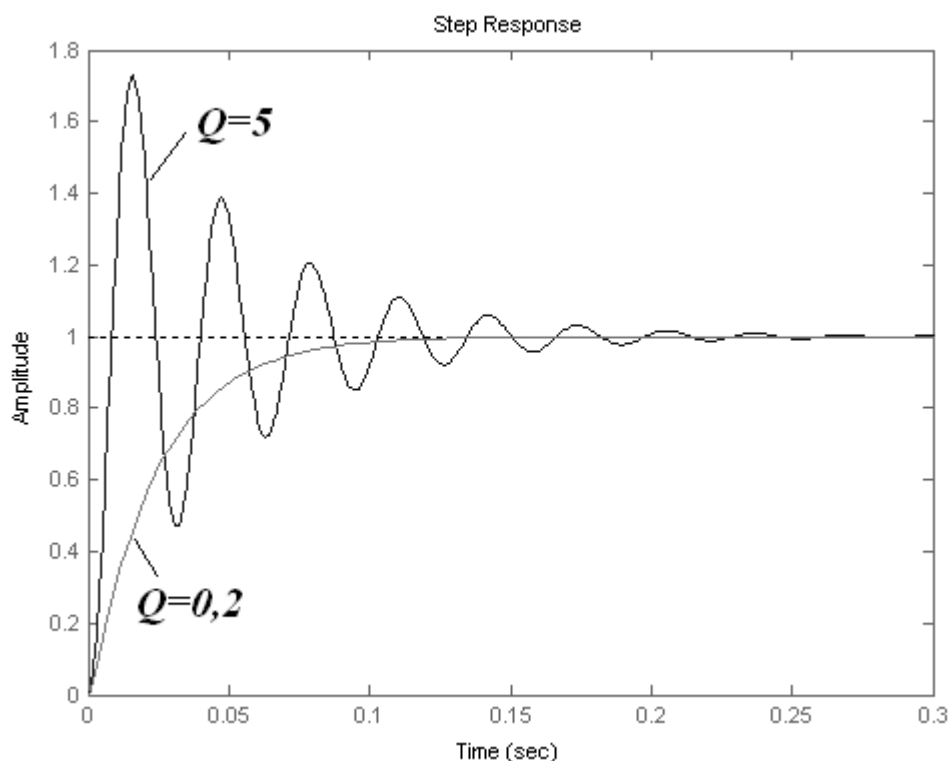


Рис. 141. Переходная характеристика активного фильтра низких частот для разных Q

Отметим также, что для активного фильтра существует область параметров R , C , при которых $Q < 0$. Это соответствует неустойчивому режиму. В этом случае реакция фильтра на любой сигнал оказывается неограниченно растущей (в линейном приближении).

Построим модель этой схемы в **SimPowerSystems**, используя созданный нами блок ОУ (**operational amplifier**), как показано на Рис. 142. В приложении модель называется **filter1_low.mdl**.

Блок операционного усилителя является нелинейным, поэтому в настройках **Simulation/Configuration Parameters Simulink** для увеличения скорости расчета нужно использовать методы **ode23tb** или **ode15s**. Также необходимо разумно выбрать шаг по времени.

Получим временные и частотные характеристики схемы для конкретных значений сопротивлений и емкостей. Пусть $R_1 = R_3 = R_6 = 100$ Ом, $R_5 = 190$ Ом, $C_2 = C_4 = 5 \cdot 10^{-5}$ Ф. Если подставить эти значения в формулы (35) и (37), то $K_p = 2,9$, $\omega_0 = 200$ рад/с и $Q = 10$.

Для случая, когда частота источника совпадает с собственной частотой системы ω_0 , сигнал на выходе фильтра достигает максимальной амплитуды (приведен на Рис. 143). Сигнал представляет собой установившиеся вынужденные колебания с частотой источника. На графике хорошо виден переходный процесс, вызванный включением схемы в момент времени $t = 0$.

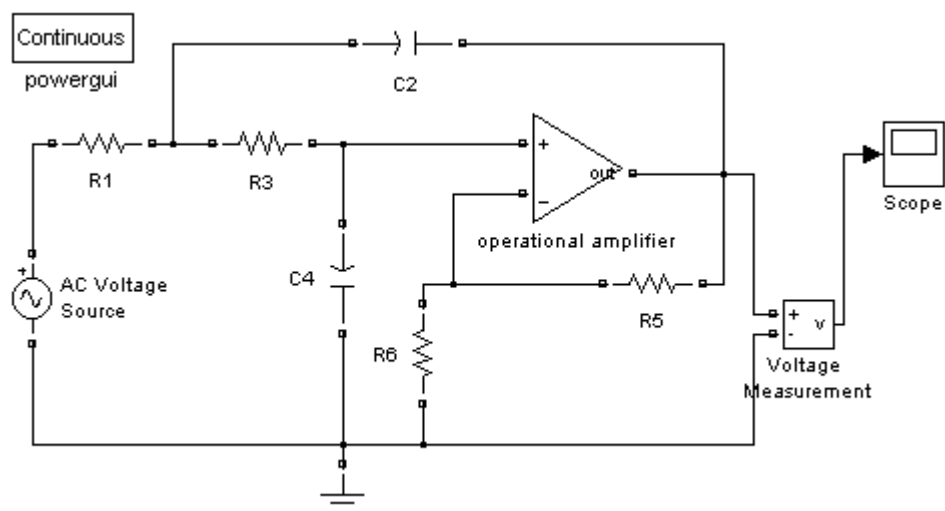


Рис. 142. Модель активного фильтра низких частот в SimPowerSystems

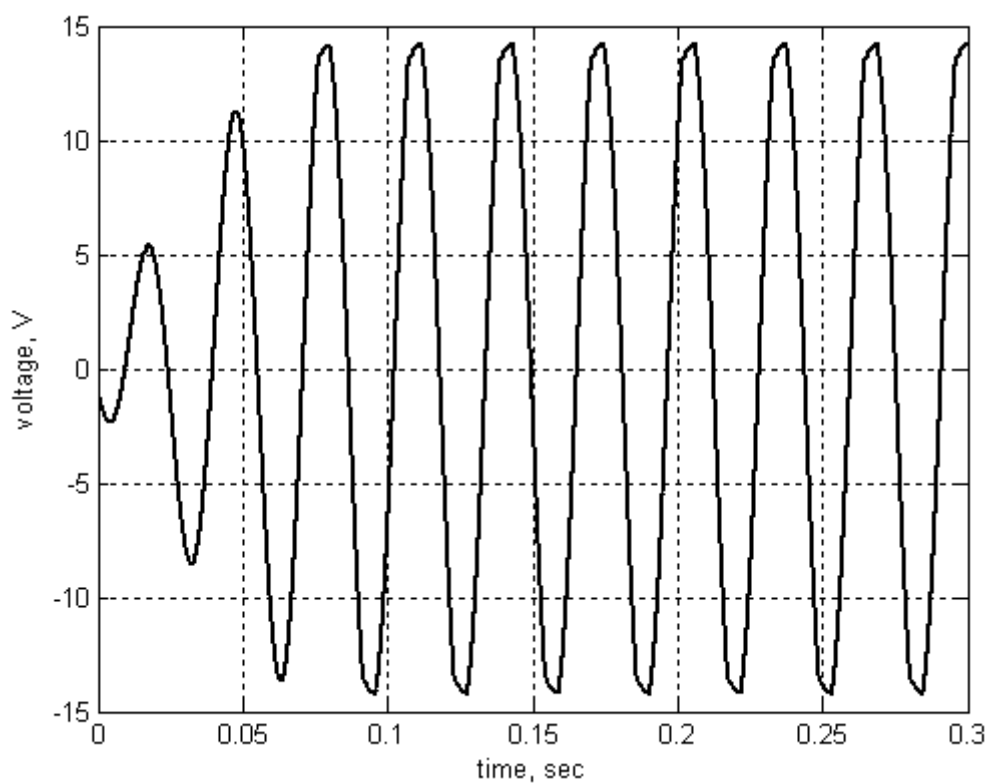


Рис. 143. Синусоидальный сигнал на выходе активного фильтра низких частот

Также на графике видны отклонения сигнала от синусоидальной формы вблизи экстремумов. На Рис. 144 приведена увеличенная часть предыдущего графика. Эти отклонения можно объяснить насыщением ОУ (максимально допустимые значения напряжения на выходе ОУ ± 15 В).

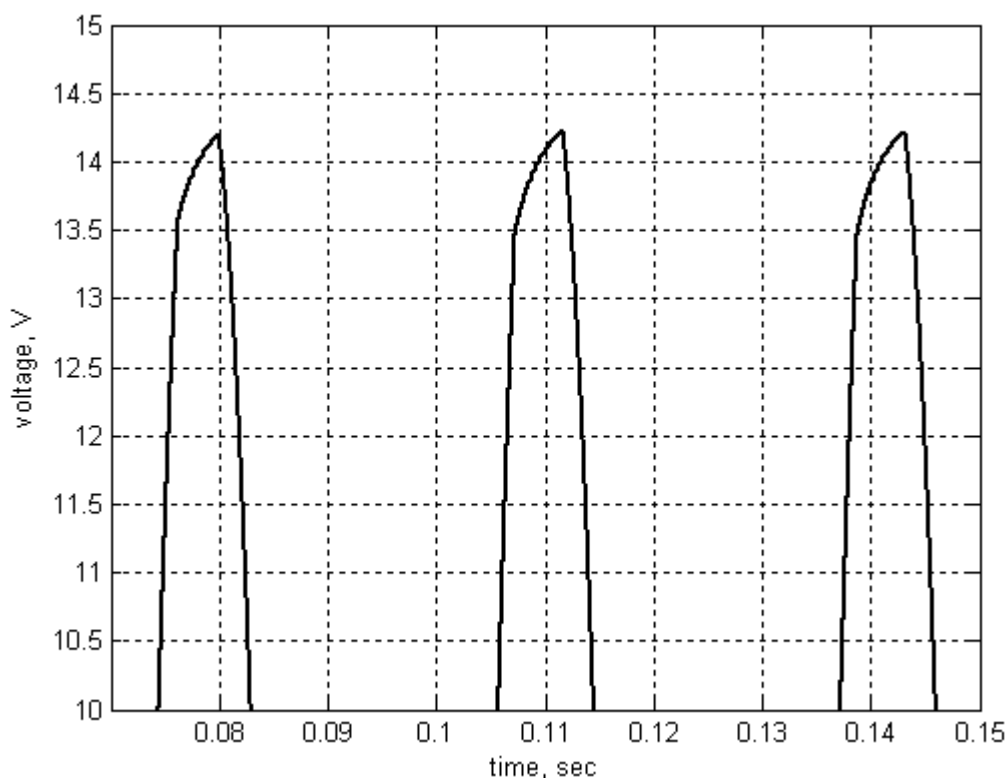


Рис. 144. Увеличение синусоидального сигнала на выходе активного фильтра низких частот

Очевидно, что при увеличении амплитуды сигнала источника увеличивается и область искажений сигнала на выходе.

Для того чтобы получить переходную характеристику этого фильтра, изменим немного SPS-модель, как показано на Рис. 145 (в приложении модель **filter1_low_step.mdl**). Здесь на вход фильтра подается единичный скачок напряжения (начало сигнала в момент времени $t = 0$) с помощью блоков **Step** и **Controlled Voltage Source** (управляемый источник напряжения).

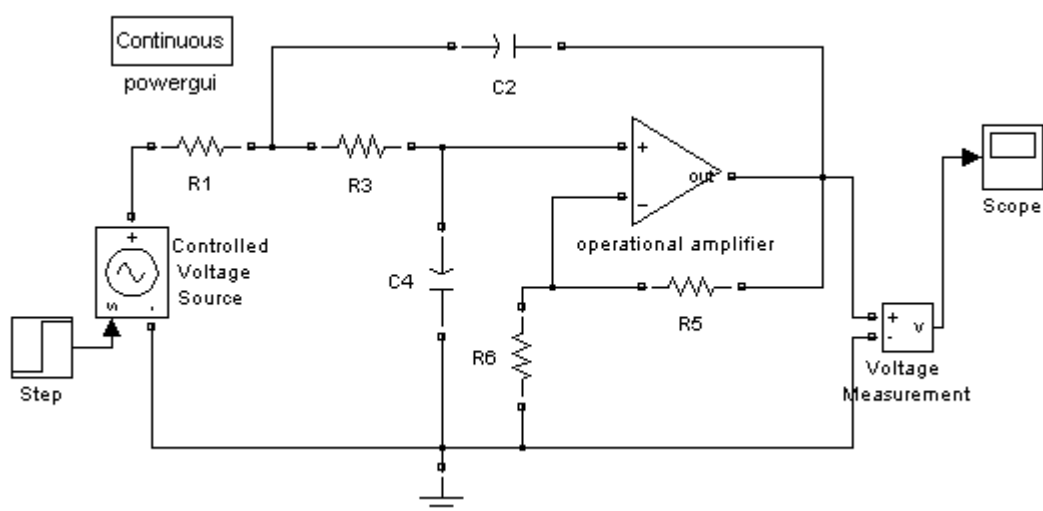


Рис. 145. Получение переходной характеристики активного фильтра низких частот

Результат работы этой схемы (переходная характеристика) представлен на Рис. 146.

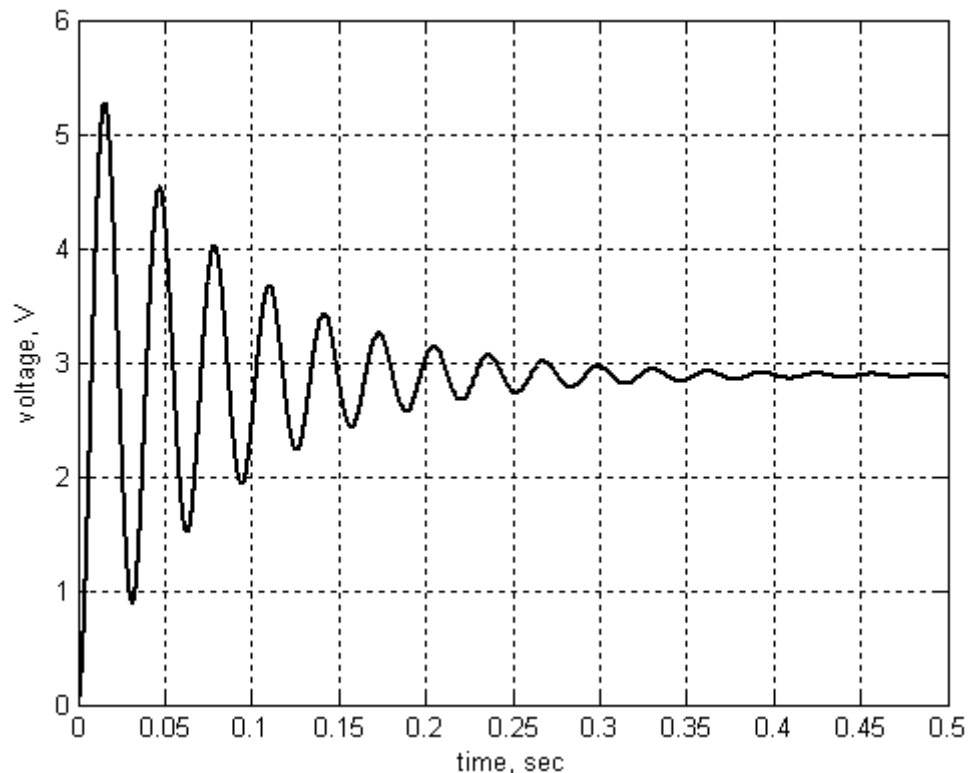


Рис. 146. Переходная характеристика активного фильтра низких частот для $Q = 10$

Для получения амплитудно-частотной характеристики этого фильтра можно на вход системы подать синусоидальный сигнал единичной амплитуды, линейно изменяющийся по частоте **chirp**. Огибающая этого сигнала на выходе системы будет АЧХ.

Мы обсудим другой способ получения АЧХ из переходной характеристики (в приложении модель **filter1_low_step2.mdl**). В схему (Рис. 145) перед осциллографом был добавлен блок **Zero-Order Hold** (Рис. 147), обеспечивающий постоянный шаг по времени. Это было сделано потому, что для получения коэффициента передачи будет использоваться дискретное преобразование Фурье (команда **MATLAB fft**), для вычисления которого требуется дискретный по времени сигнал, отсчёты которого сделаны через равные промежутки времени. В настройках блока **Zero-Order Hold** задаем параметр **Sample time** (время квантования) равный $1/2000$, где 2000 это частота квантования в герцах f_s (частоту квантования следует выбирать, согласуясь с теоремой Котельникова и возможностями компьютера).

Идея метода заключается в том, что коэффициент передачи линейной системы одновременно является и спектром импульсной характеристики, а импульсная характеристика может быть получена как производная переходной характеристики. Поэтому, чтобы получить коэффициент передачи, мож-

но записать переходную характеристику в рабочую область **MATLAB** и выполнить преобразование Фурье производной переходной характеристики.

В настройках решателя **Simulink** (**Simulation/Configuration Parameters/Type**) используется алгоритм решения с переменным шагом по времени **Variable-step**, а для дискретного преобразования Фурье требуется постоянный шаг по времени. Поэтому, как уже говорилось, перед блоком **Scope**, сохраняющим сигнал в рабочей области, поставлен блок **Zero-Order Hold**, в котором указан шаг по времени. Этот блок не нужно использовать, если в настройках решателя выбрать фиксированный по времени шаг **Fixed-step**.

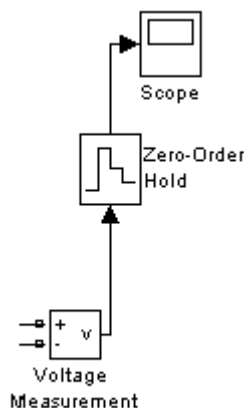


Рис. 147. Изменение схемы активного фильтра низких частот для получения АЧХ

После проведения моделирования в **Simulink** в рабочей области **MATLAB** в переменной **ScopeData** (при соответствующих настройках блока «Осциллограф») будет записана переходная характеристика фильтра. Переменная **ScopeData** является структурой, имеющей сложное строение. В этой переменной отсчеты сигнала и соответствующие им отсчеты времени хранятся отдельно. Чтобы получить вектор временных отсчетов используется переменная **ScopeData.time**, а вектор отсчетов сигнала **ScopeData.signals.values**.

Например, чтобы построить график зависимости сигнала от времени, используется команда **plot (ScopeData.time, ScopeData.signals.values)**.

Рассмотрим листинг программы построения АЧХ в **MATLAB**.

```
fs=2000; %частота квантования
%сохраняем сигнал в новую переменную:
y=ScopeData.signals.values;
%дифференцируем переходную характеристику
%и получаем импульсную:
g=diff(y)*fs;
N=length(g); %длина вектора g
plot(g)
%быстрое преобразование Фурье от импульсной
%характеристики:
```



```

K=fft(g)/fs;           %передаточная функция
w=2*pi*(0:N-1)*fs/N;   %создаем вектор частот
plot(w, abs(K))         %АЧХ

```

Амплитуду k -ой гармоники c_k можно выразить как отношение результата дискретного преобразования Фурье к количеству отсчетов N или как отношение коэффициента передачи K к длительности записи сигнала T :

$$c_k = \frac{X_k}{N} = \frac{K(\omega_k)}{T}$$

Отсюда, учитывая, что $T = t_s N = \frac{N}{f_s}$, где t_s – время квантования (**Sample time**), отсчеты передаточной функции будут:

$$K(\omega_k) = \frac{X_k T}{N} = \frac{X_k}{f_s}.$$

И отсчеты частоты будут:

$$\omega_k = \frac{2\pi k}{T} = \frac{2\pi k f_s}{N}.$$

АЧХ этого фильтра представлена на Рис. 148.

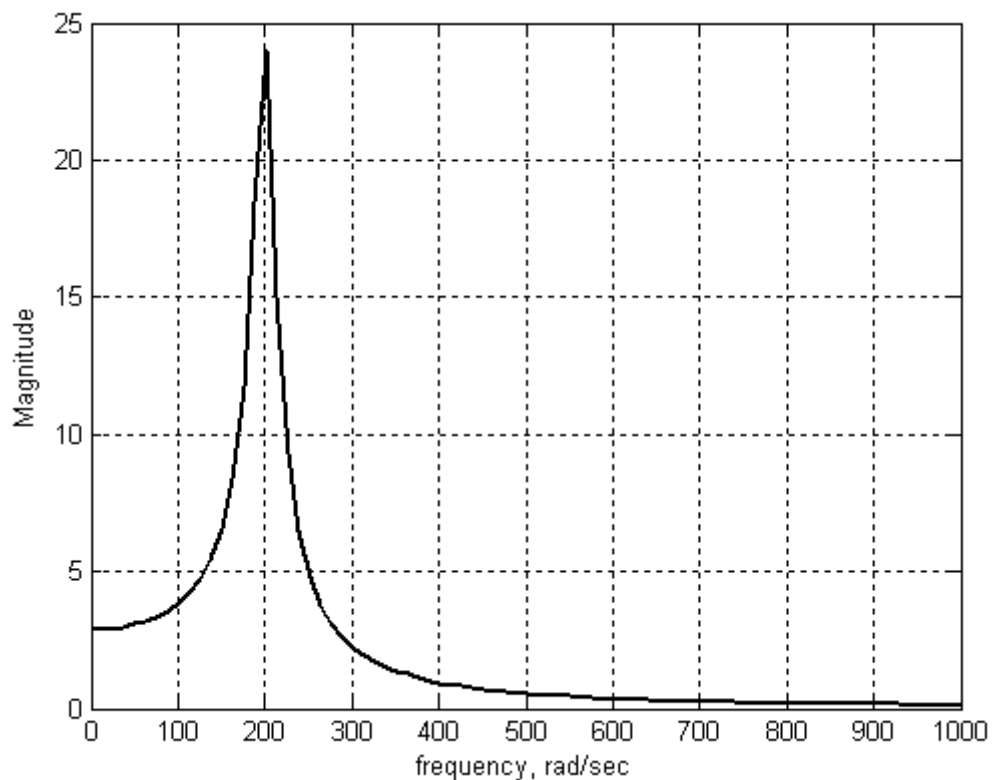


Рис. 148. АЧХ активного фильтра низких частот

5.1.5. Активный RC-фильтр высоких частот второго порядка

Для реализации этого фильтра в схеме, представленной на Рис. 135, параметры нужно выбрать так: $Z_1 = 1/sC_1$, $Z_2 = R_2$, $Z_3 = 1/sC_3$, $Z_4 = R_4$. Конкретная схема приведена на Рис. 149.

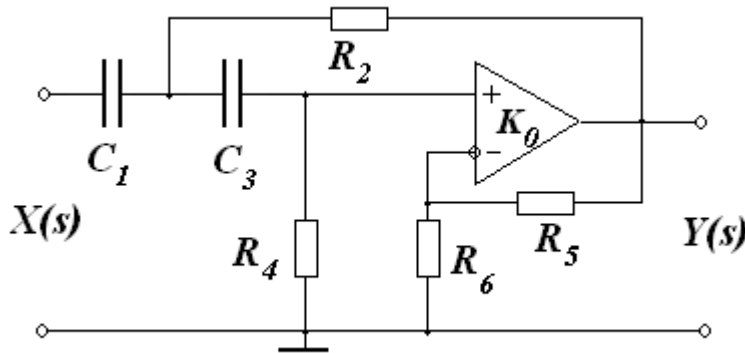


Рис. 149. Активный RC-фильтр высоких частот второго порядка

Также прямой подстановкой значений указанных параметров в формулу (33) получаем следующее выражение для коэффициента передачи:

$$K(s) = K_p \frac{s^2}{s^2 + \frac{\omega_0}{Q}s + \omega_0^2}, \quad (44)$$

где при $K_0 \gg 1$

$$\omega_0^2 = \frac{1}{C_1 C_3 R_2 R_4}, \quad K_p \approx \frac{R_5 + R_6}{R_6},$$

$$Q = \frac{\sqrt{\frac{R_4 C_1}{R_2 C_3}}}{1 + \frac{C_1}{C_3} - \frac{R_4 R_5}{R_6 R_2}} \approx \frac{\sqrt{\frac{R_4 C_1}{R_2 C_3}}}{1 + \frac{C_1}{C_3} + \frac{R_4}{R_2} (1 - K_p)}.$$

В этом фильтре выбором R и C также можно обеспечить вещественные ($Q < 0,5$) и комплексные ($Q > 0,5$) значения корней полинома в знаменателе коэффициента передачи.

Исследовать АЧХ и ФЧХ рассматриваемого фильтра можно также, как и в предыдущем случае (п. 5.1.4.) в **Simulink**. Для параметров электрической схемы $K_p = 1$, $\omega_0 = 200$ рад/с вектор числителя передаточной функции будет $b = [1 \ 0 \ 0]$ и вектор знаменателя $a = [1 \ 2e2/Q \ 4e4]$.

Рассмотрим другой способ описания этой системы с помощью библиотеки **Control System**. В **MATLAB** поддерживается объектно-ориентированное программирование, и эта библиотека как раз предоставляет набор объектов, с помощью которых удобно исследовать линейные системы.

Например, в этой библиотеке находится **LTI Viewer**, которым мы уже пользовались, не вникая к какой библиотеке **Simulink** он принадлежит. Этот графический интерфейс также может быть вызван и из командного окна **MATLAB** или с помощью m-файла командой **ltiview**.

Линейная система в данном случае задаётся с помощью передаточной функции. В **Control System Toolbox** передаточная функция создаётся командой **tf(num,den)**, где **num** – вектор с коэффициентами числителя полинома передаточной функции, **den** – вектор с коэффициентами полинома в знаменателе.

В библиотеке **Control System Toolbox** находится большой набор функций для исследования линейных систем. Например, с помощью функции **freqresp** получаются значения АЧХ и ФЧХ для заданных значений частот. Первый аргумент этой команды – это передаточная функция, второй – вектор со значениями частоты в рад/с. Функция возвращает трёхмерный массив, первый индекс массива соответствует номеру входа линейной системы, второй индекс – номеру выхода, третий индекс номеру элемента вектора со значениями частоты. В данном примере линейная система имеет один вход и один выход.

Для построения графиков данные из трёхмерного массива необходимо сохранить в одномерном массиве. Команда **plot** позволяет строить сразу несколько графиков. Для этого данные должны быть сохранены в матрице, каждая строка матрицы соответствует одному графику, так как вектор f – вектор-строка.

Рассмотрим листинг команд в **MATLAB** (в приложении файл **filter1_high_mat.m**):

```
w0 = 200; % резонансная частота (рад/с)
% диапазон частот в Гц в логарифмическом масштабе:
f = logspace(0,3,1000);

% создаём три передаточных функции:
h1 = tf([1 0 0], [1 w0/0.2 w0^2]); % Q = 0.2
h2 = tf([1 0 0], [1 w0/1 w0^2]); % Q = 1
h3 = tf([1 0 0], [1 w0/3 w0^2]); % Q = 3

% рассчитываем частотные характеристики:
R1 = freqresp(h1, 2*pi*f);
R2 = freqresp(h2, 2*pi*f);
R3 = freqresp(h3, 2*pi*f);

% изменяем размерность, чтобы построить графики:
H(1:3,1:1000)=[R1(1,1,:);R2(1,1,:);R3(1,1,:)];

% строим графики АЧХ и ФЧХ:
subplot(2,1,1), semilogx(f,abs(H)),
```

```

legend('Q=0.2','Q=1','Q=3')
subplot(2,1,2), semilogx(f, angle(H))

```

Мы создали три объекта передаточной функции $h1$, $h2$, $h3$ командой **tf** с разными значениями добротности Q (0,2, 1 и 3 соответственно). Для каждой передаточной функции мы рассчитали ее частотные характеристики с помощью команды **freqresp** и записали их в переменные $R1$, $R2$, $R3$. Выделили из этих переменных информацию о частотных характеристиках и записали ее в массив H . Массив H состоит из трех строк с набором значений коэффициента передачи (комплексные величины) для частот $2\pi f$ для каждой добротности Q .

Зависимость частотных характеристик от величины параметра Q показаны на Рис. 150.

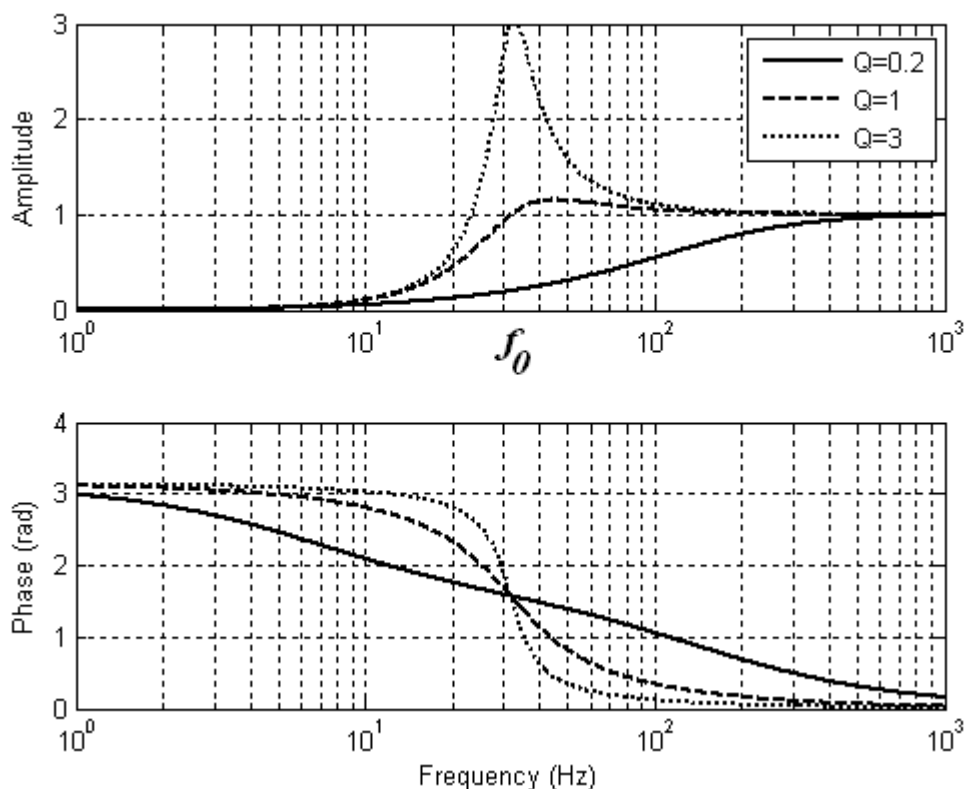


Рис. 150. АЧХ и ФЧХ активного RC-фильтра высоких частот второго порядка

Собственная частота системы в герцах $f_0 = 31.8310$. Здесь также при больших значениях Q ($Q \gg 1$) АЧХ активного фильтра имеет резонансный характер. Схема фильтра высоких частот второго порядка на RLC -элементах с аналогичной частотной характеристикой приведена на Рис. 151. Коэффициент передачи такого фильтра описывается выражением (44) со следующими значениями параметров:

$$\omega_0 = \frac{1}{LC}, \quad Q = \frac{\omega_0 L}{R}, \quad K_p = 1.$$

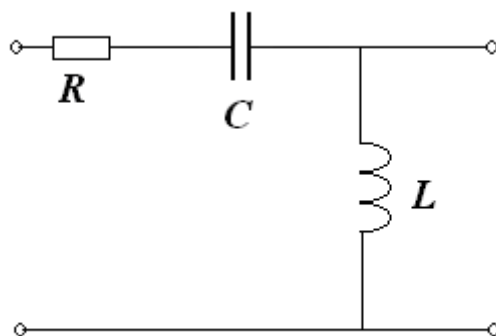


Рис. 151. Схема фильтра высоких частот второго порядка на RLC -элементах

Аналогично предыдущему варианту фильтра существует область значений R, C при которых $Q < 0$ и фильтр высоких частот также становится неустойчивым.

Определим переходную характеристику фильтра, учитывая, что корни знаменателя его коэффициента передачи определяются выражением (38):

$$h(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{K_p s}{(s-s_1)(s-s_2)} e^{st} ds$$

Разлагая дробь, входящую в подынтегральное выражение для $h(t)$, на простейшие

$$\frac{s}{(s-s_1)(s-s_2)} = \frac{s_1}{s_1-s_2} \frac{1}{s-s_1} + \frac{s_2}{s_2-s_1} \frac{1}{s-s_2},$$

и учитывая (42), получим

$$\begin{aligned} h(t) &= \frac{K_p s_1}{s_1-s_2} \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{(s-s_1)} e^{st} ds + \frac{K_p s_2}{s_2-s_1} \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{(s-s_2)} e^{st} ds = \\ &= \frac{K_p s_1}{s_1-s_2} e^{s_1 t} + \frac{K_p s_2}{s_2-s_1} e^{s_2 t} \end{aligned} \quad (45)$$

Семейство переходных характеристик при различных значениях параметра Q (для рассмотренного выше примера) представлено на Рис. 152.

Этот график можно получить, добавив к приведенному выше листингу команд MATLAB еще две команды:

```
figure %открыть новое окно для графика
step(h1, h2, h3) %рассчитать и построить
%переходную характеристику функций h1, h2, h3
```

Команда **step** строит переходную характеристику линейной модели (модель должна быть заранее создана командой **TF** – передаточная функция (или с помощью других представлений линейной системы: **ZPK** – нули-полюса или **SS** – пространство состояний)). Если в команде специально не указано (как в нашем случае), то диапазон времени и число точек для построения графиков выбирается автоматически.

В **Control System Toolbox** также находятся команды **impulse** для получения импульсной характеристики линейной системы, **lsim(SYS,U,T)** – для получения сигнала во временной области линейной системы **SYS** при прохождении через нее произвольного сигнала, который задается векторами **U** и **T**. Также можно получить АЧХ и ФЧХ системы командой **bode** и использовать другие возможности **LTI-viewer**, или вызвать графический интерфейс пользователя этого инструмента командой **ltiview**. Подробнее с этими возможностями можно ознакомиться в справке **MATLAB**.

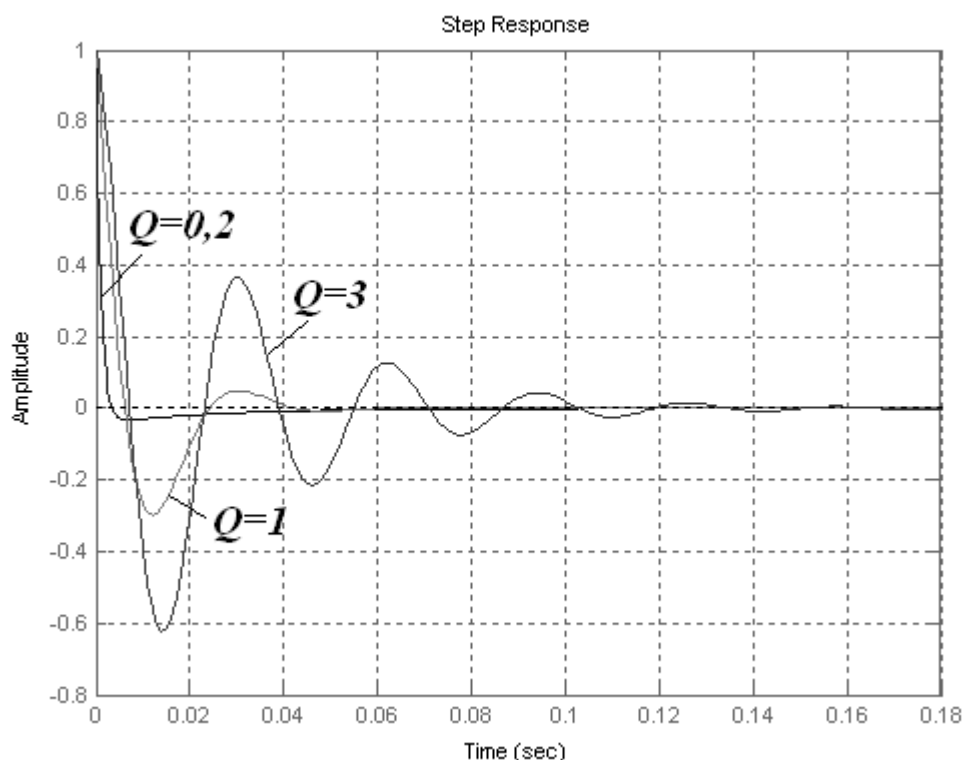


Рис. 152. Переходные характеристики активного RC-фильтра высоких частот второго порядка

5.1.6. Режекторный активный RC-фильтр

Такой фильтр (Рис. 153) с регулируемой полосой режекции может быть построен по несколько измененной первой базовой схеме (Рис. 135). Имя модели этого фильтра в приложении **filter1_reject.mdl**.

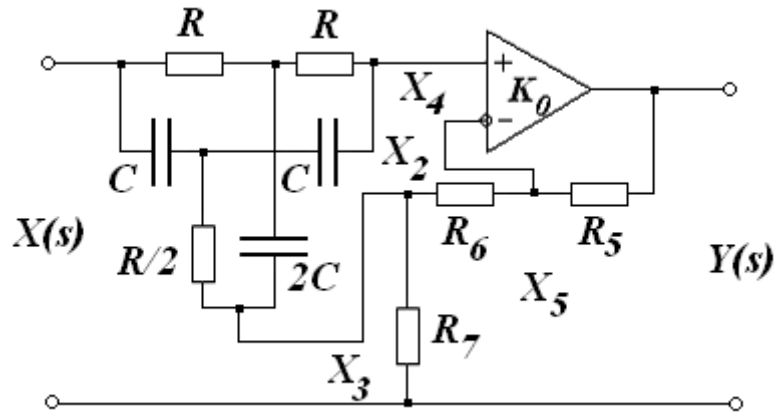


Рис. 153. Режекторный активный RC -фильтр

Изменение заключается в том, что обе цепи обратной связи имеют общую часть. В цепи частотноизбирательной обратной связи в этом случае включен сбалансированный двойной Т-образный RC -мост (Рис. 154).

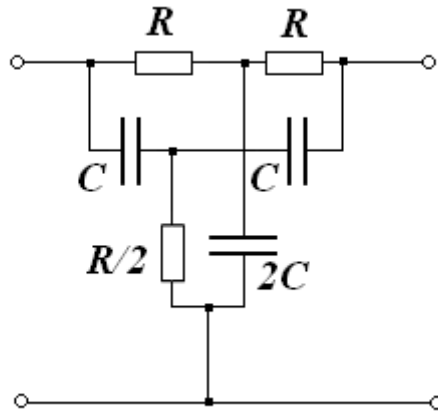


Рис. 154. Сбалансированный двойной Т-образный RC -мост

Коэффициент передачи Т-образного RC -моста в предположении, что мост не нагружен, равен

$$\beta(s) = \frac{s^2 + \omega_0^2}{s^2 + 4\omega_0 s + \omega_0^2}, \quad \omega_0 = \frac{1}{RC}. \quad (46)$$

Такое приближение может быть принято, т.к. входное сопротивление усилителя велико. Для определения коэффициента передачи используем соотношения (Рис. 153):

$$\begin{aligned} X_3(s) &= \frac{R_7}{R_5 + R_6 + R_7} Y(s), \quad X_5(s) = \frac{R_6 + R_7}{R_5 + R_6 + R_7} Y(s), \\ Y(s) &= K_0 X_4(s) = K_0 (X_2(s) + X_3(s) - X_5(s)) = \\ &= K_0 [\beta(s)(X(s) - X_3(s)) + X_3(s) - X_5(s)] = \\ &= K_0 [\beta(s)X(s) + (1 - \beta(s))X_3(s) - X_5(s)] \end{aligned}$$

Подставляя в это выражение значения X_3 и X_5 , находим

$$K(s) = \frac{Y(s)}{X(s)} = \frac{K_0 \beta(s)}{1 + \frac{\beta(s) R_7 + R_6}{R_5 + R_6 + R_7} K_0}.$$

Выделим теперь выражения, описывающие величину усиления, регулируемое соотношением сопротивлений в цепи обратной связи:

$$K_1 = \frac{K_0}{1 + \frac{R_6}{R_5 + R_6 + R_7} K_0}, \quad \alpha = \frac{R_7}{R_5 + R_6 + R_7}.$$

Легко убедиться, что тогда

$$K(s) = \frac{K_1 \beta(s)}{1 + K_1 \alpha \beta(s)}.$$

Используя выражение (46) для $\beta(s)$, получим

$$K(s) = \frac{K_1 (s^2 + \omega_0^2)}{(K_1 \alpha + 1) \left[s^2 + 4 \frac{\omega_0}{(K_1 \alpha + 1)} s + \omega_0^2 \right]}.$$

Если $\frac{R_6}{R_5 + R_6 + R_7} K_0 \gg 1$, то $K_1 \alpha + 1 \approx \frac{R_6 + R_7}{R_6}$. Таким образом, в этом случае коэффициент передачи фильтра не зависит от K_0 и определяется выражением

$$K(s) = \left(1 + \frac{R_5}{R_6 + R_7} \right) \frac{s^2 + \omega_0^2}{s^2 + 4 \omega_0 \frac{R_6}{R_6 + R_7} s + \omega_0^2}. \quad (47)$$

Введя обозначения

$$K_r = 1 + \frac{R_5}{R_6 + R_7}, \quad Q = \frac{R_6 + R_7}{4R_6},$$

приведем выражение (47) к стандартному виду

$$K(s) = K_r \frac{s^2 + \omega_0^2}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2}.$$

Семейство АЧХ и ФЧХ активного режекторного RC-фильтра при различных значениях Q приведены на (Рис. 155). Здесь для определенности выбраны значения $K_r = 10$, $\omega_0 = 100$ рад/с.

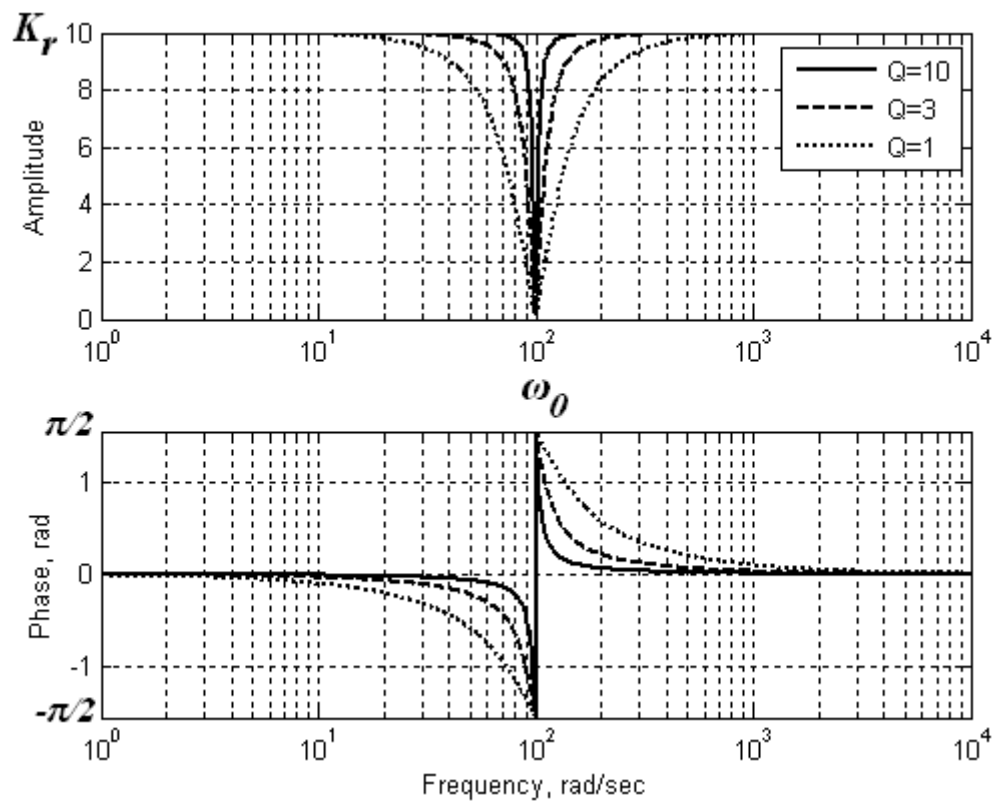


Рис. 155. АЧХ и ФЧХ активного режекторного RC-фильтра

Полоса режекции определяется величиной Q и оказывается тем более узкой, чем больше Q . Существенно, что при любых значениях K_r режекторный активный фильтр остается устойчивым, т.к. всегда Q больше нуля.

Пример программы в **MATLAB** для получения этих характеристик приведен ниже (в приложении программа **filter1_reject_mat.m**). Здесь используются уже известные нам команды **tf** и **freqresp**.

```
w=logspace(0, 4, 1000);           %диапазон частот

Kr=10;
w0=100;

Q=10;
b=[Kr 0 Kr*w0^2];
a=[1 w0/Q w0^2];
h1 = freqresp(tf(b,a),w);

Q=3;
a=[1 w0/Q w0^2];
h2 = freqresp(tf(b,a),w);

Q=1;
a=[1 w0/Q w0^2];
```

```
h3 = freqresp(tf(b,a),w);
```

```
H(1:3,:) = [h1(1,1,:); h2(1,1,:); h3(1,1,:)];
subplot(2,1,1),
semilogx(w, abs(H)), legend('Q=10','Q=3','Q=1')
subplot(2,1,2), semilogx(w, angle(H))
```

Модель активного режекторного *RC*-фильтра в *SPS* будет выглядеть, как показано на Рис. 156. Воспользуемся рассмотренным в предыдущем параграфе методом нахождения АЧХ из передаточной функции для несбалансированного моста (модель в приложении **filter1_reject_unbalanced.mdl**).

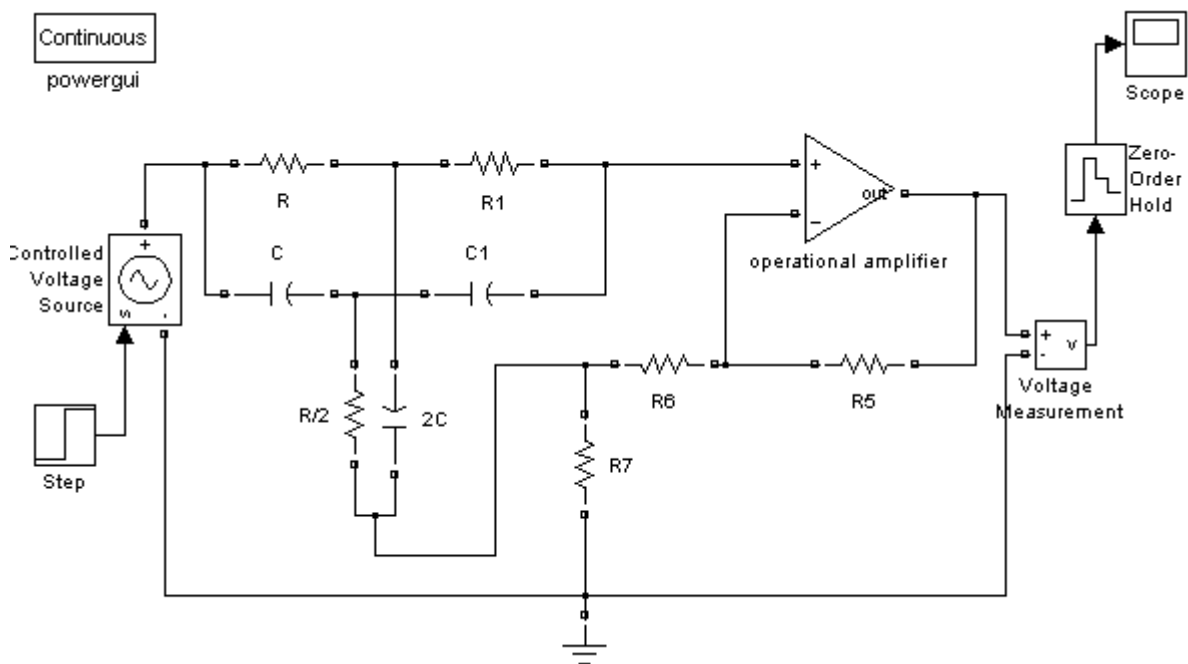


Рис. 156. Модель активного режекторного *RC*-фильтра в *SPS*

К уже известной нам программе (получение АЧХ в п. 5.1.4.) в **MATLAB** добавится команда построения ФЧХ **angle(K)** (аргумент от передаточной функции).

```
fs=2*pi*1000; %частота квантования
y=ScopeData.signals.values;
g=diff(y)*fs;
N=length(g);
K=fft(g)/fs; %передаточная функция
w=2*pi*(0:N-1)*fs/N; %создаем вектор частот
subplot(2,1,1), semilogx(w, abs(K)) %АЧХ
subplot(2,1,2), semilogx(w, angle(K)) %ФЧХ
```

Выберем параметры электрической схемы. Мы описывали эту схему в предположении, что Т-образный мост не нагружен. Поэтому, для того чтобы

результаты моделирования согласовывались с теоретическими результатами, выходное сопротивление Т-образного моста должно быть много меньше входного сопротивления усилителя и одновременно много больше сопротивления R_7 , через которое средняя точка моста подключена к земле. Выберем параметры:

$$R=10^4 \text{ Ом}, C=10^{-6} \text{ Ф.}$$

Тогда для $Q=1$, $K_r=10$ и $\omega_0=100$ рад/с

$$R_6=100 \text{ Ом},$$

$$R_5=3600 \text{ Ом},$$

$$R_7=300 \text{ Ом}.$$

Если мост не сбалансирован (не точно выполняются соотношения между R и C в вертикальных и горизонтальных плечах моста (Рис. 154)), то режекция не является полной.

На Рис. 157 изображены характеристики для сбалансированного (пунктирная линия) и несбалансированного (сплошная линия) моста.

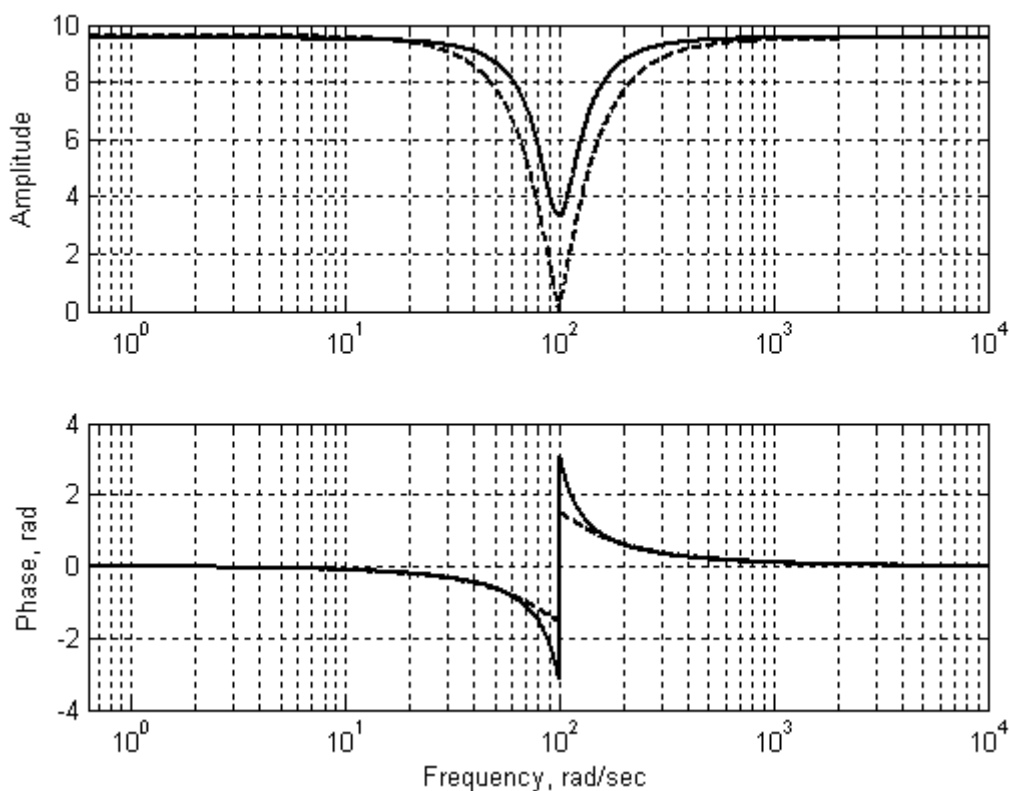


Рис. 157. АЧХ и ФЧХ для сбалансированного и несбалансированного моста

Определим реакцию режекторного фильтра на включение синусоидального сигнала единичной амплитуды, частота которого совпадает с частотой режекции фильтра ω_0 . Для такого сигнала

$$\begin{aligned}
X(s) &= \int_0^{\infty} \sin \omega_0 t e^{-st} dt = \frac{1}{2i} \left[\int_0^{\infty} e^{(i\omega_0 - s)t} dt - \int_0^{\infty} e^{(-i\omega_0 - s)t} dt \right] = \\
&= \frac{1}{2i} \left[\frac{1}{s - i\omega_0} - \frac{1}{s + i\omega_0} \right] = \frac{\omega_0}{s^2 + \omega_0^2}
\end{aligned}$$

Сигнал на выходе фильтра

$$\begin{aligned}
y(t) &= \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{s^2 + \omega_0^2}{(s - s_1)(s - s_2)} \frac{\omega_0}{s^2 + \omega_0^2} e^{st} ds = \\
&= \frac{\omega_0}{s_1 - s_2} \left[\frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{(s - s_1)} e^{st} ds - \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{(s - s_2)} e^{st} ds \right] = \\
&= \frac{\omega_0}{s_1 - s_2} [e^{s_1 t} - e^{s_2 t}]
\end{aligned} \tag{48}$$

Корни знаменателя коэффициента передачи s_1, s_2 по-прежнему определяются выражением (38). При $Q < 1/2$, т.е. при $R_7 < R_6$, оба корня вещественны и отрицательны. Выходной сигнал, соответствующий этому случаю, изображен на Рис. 158.

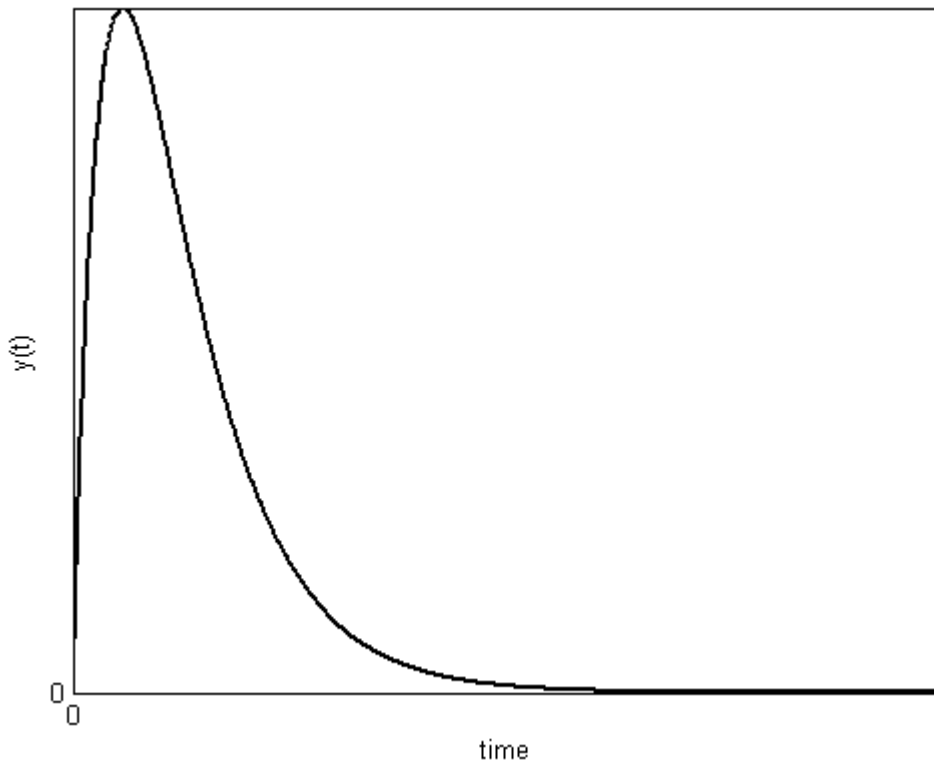


Рис. 158. Реакция режекторного фильтра на включение синусоидального сигнала единичной амплитуды: случай, когда $Q < 1/2$

При $Q > 1/2$, корни s_1 и s_2 будут комплексно-сопряженными. Согласно (38), представим их в следующем виде:

$$s_{1,2} = -\frac{\omega_0}{2Q} \pm i\omega_0 \sqrt{1 - \frac{1}{4Q^2}}.$$

С учетом этого выражение для выходного сигнала примет вид

$$y(t) = \frac{1}{\sqrt{1 - \frac{1}{4Q^2}}} \exp\left(-\frac{\omega_0}{2Q}t\right) \sin\left(\omega_0 \sqrt{1 - \frac{1}{4Q^2}}t\right).$$

Таким образом, переходной процесс имеет характер затухающих колебаний. Причем с ростом коэффициента Q затухание уменьшается, а частота осцилляций приближается к частоте резонанса (Рис. 159).

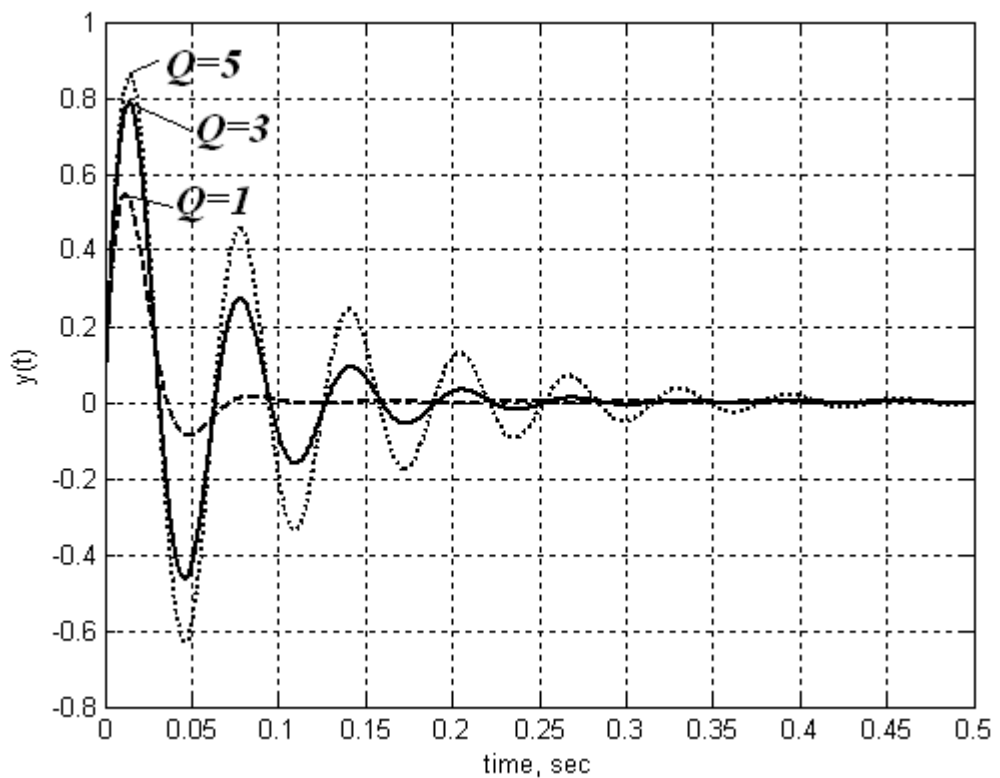


Рис. 159. Реакция режекторного фильтра на включение синусоидального сигнала единичной амплитуды для разных Q

5.1.7. Основные типы активных фильтров: схема второго типа

Рассмотрим теперь схемы активных фильтров второго типа, в которых частотнозависимая обратная связь подается на инвертирующий вход ОУ, а частотнезависимая – на неинвертирующий вход. Блок-схема такого фильтра приведена на Рис. 160.

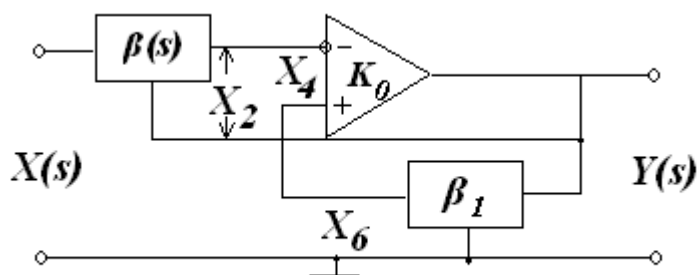


Рис. 160. Блок-схема фильтра второго типа

Легко убедиться, что коэффициент передачи этой схемы отличается от (26) лишь изменением знака у K_0 . Поэтому

$$K(s) = \frac{-\beta(s)K_0}{1 - \beta_1 K_0 - [\beta(s) - 1]K_0} = \frac{-K_p \beta(s)}{1 - K_p [\beta(s) - 1]},$$

где теперь

$$K_p = \frac{K_0}{1 - \beta_1 K_0}.$$

Обычно не требуется увеличивать усиление ($K_p > K_0$), поэтому на практике положительную обратную связь не включают. Конкретный пример реализации схемы второго типа приведен на Рис. 161.

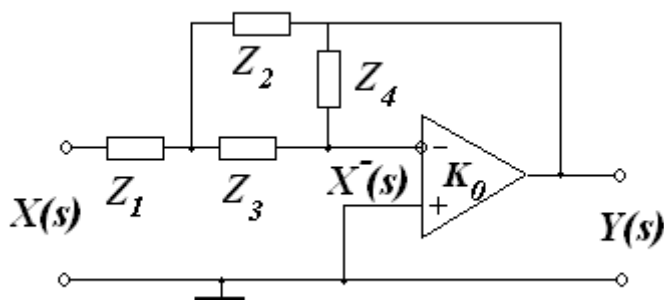


Рис. 161. Пример реализации схемы второго типа

Для определения искомого коэффициента передачи и в этом случае удобно рассмотреть эквивалентную схему с двумя генераторами напряжения (Рис. 162).

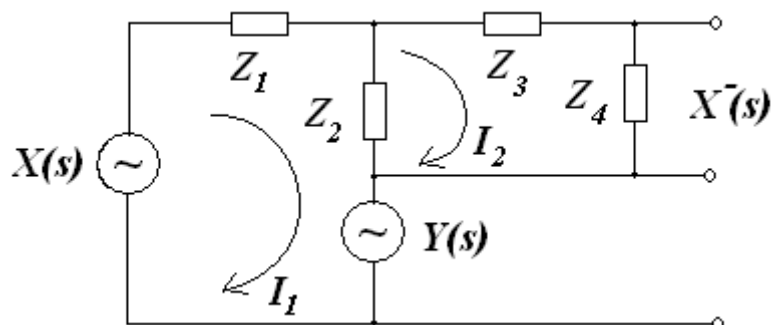


Рис. 162. Эквивалентная схема

Система уравнений Кирхгофа при выбранных направлениях токов и напряжений запишется следующим образом:

$$\begin{aligned} X(s) - Y(s) &= (Z_1 + Z_2)I_1(s) - Z_2I_2(s), \\ 0 &= -Z_2I_1(s) + (Z_2 + Z_3 + Z_4)I_2(s) \end{aligned} \quad (49)$$

Определитель системы:

$$\Delta = (Z_1 + Z_2)(Z_3 + Z_4) + Z_1Z_2$$

Найдем $I_2(s)$ из системы уравнений (49):

$$I_2(s) = \frac{X(s) - Y(s)}{\Delta} Z_2.$$

Из схем, приведенных на Рис. 161 и Рис. 162, следует, что

$$X^-(s) = -\frac{Y(s)}{K_0} = Y(s) + Z_4I_2(s),$$

отсюда

$$-Y(s) \left(1 + \frac{1}{K_0} - \frac{Z_2 Z_4}{\Delta} \right) = \frac{X(s) Z_2 Z_4}{\Delta}.$$

Таким образом, искомый коэффициент передачи будет равен

$$K(s) = \frac{-Z_2 Z_4}{\Delta \left(1 + \frac{1}{K_0} - \frac{Z_2 Z_4}{\Delta} \right)}. \quad (50)$$

При $K_0 \gg 1$

$$K(s) \approx \frac{Z_2 Z_4}{Z_2 Z_4 - \Delta}. \quad (51)$$

Рассмотрим ряд конкретных примеров активных фильтров второго типа.

5.1.8. Примеры активных полосопропускающих фильтров второго типа

Пример 1.

В схеме на Рис. 161 положим $Z_1 = R_1$, $Z_2 = 1/sC_1$, $Z_3 = 1/sC_2$, $Z_4 = R_2$. В результате получим схему, изображенную на Рис. 163.

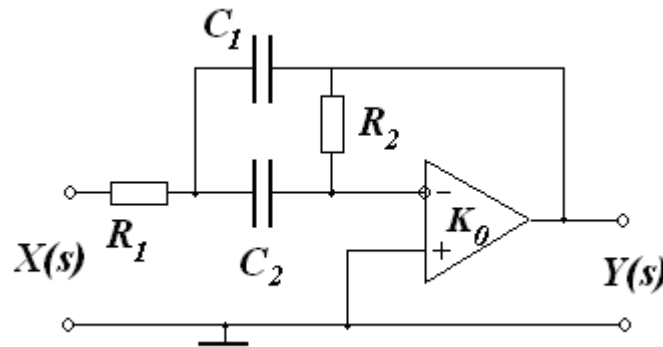


Рис. 163. Активный полосопропускающий фильтр второго типа

Подстановка значений параметров Z_k в (51) дает следующее выражение для коэффициента передачи:

$$K(s) = -\frac{K_1 s}{s^2 + \frac{\omega_0}{Q} s + \omega_0^2}, \quad (52)$$

где

$$K_1 = \frac{1}{R_1 C_1}, \quad Q = \sqrt{\frac{R_2}{R_1}} \frac{\sqrt{C_1 C_2}}{C_1 + C_2}, \quad \omega_0^2 = \frac{1}{R_1 R_2 C_1 C_2} \quad (53)$$

Пример 2.

Фильтр с характеристикой, подобной (52), можно реализовать иначе. Пусть теперь элементы схемы Рис. 161 имеют значения $Z_1 = 1/sC_1$, $Z_2 = R_2$, $Z_3 = R_3$, $Z_4 = 1/sC_4$ (Рис. 164).

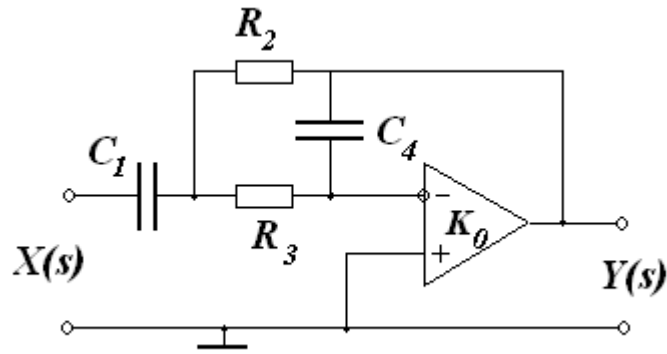


Рис. 164. Активный полосопропускающий фильтр второго типа

Подставляя Z_k в (51), получим выражение для коэффициента передачи, совпадающее с (52), где теперь

$$K_1 = \frac{1}{R_3 C_4}, \quad Q = \sqrt{\frac{R_3 C_1}{R_2 C_4}} \frac{1}{1 + R_3/R_2}, \quad \omega_0^2 = \frac{1}{R_3 R_2 C_1 C_4} \quad (54)$$

Выражение (52) при $K_1 = 1$ совпадает с коэффициентом передачи RLC -фильтра, собранного по схеме (Рис. 165), со следующими параметрами:

$$Q = \sqrt{\frac{L}{C}} \frac{1}{R}, \quad \omega_0 = \frac{1}{\sqrt{LC}} \quad (55)$$

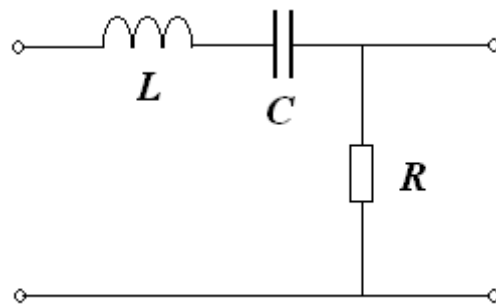


Рис. 165. Аналогичный RLC -фильтр

АЧХ и ФЧХ рассматриваемого активного полосопропускающего фильтра при разных Q приведены на (Рис. 166). Графики приведены для $K_I = 10$ и $\omega_0 = 100$ рад/с.

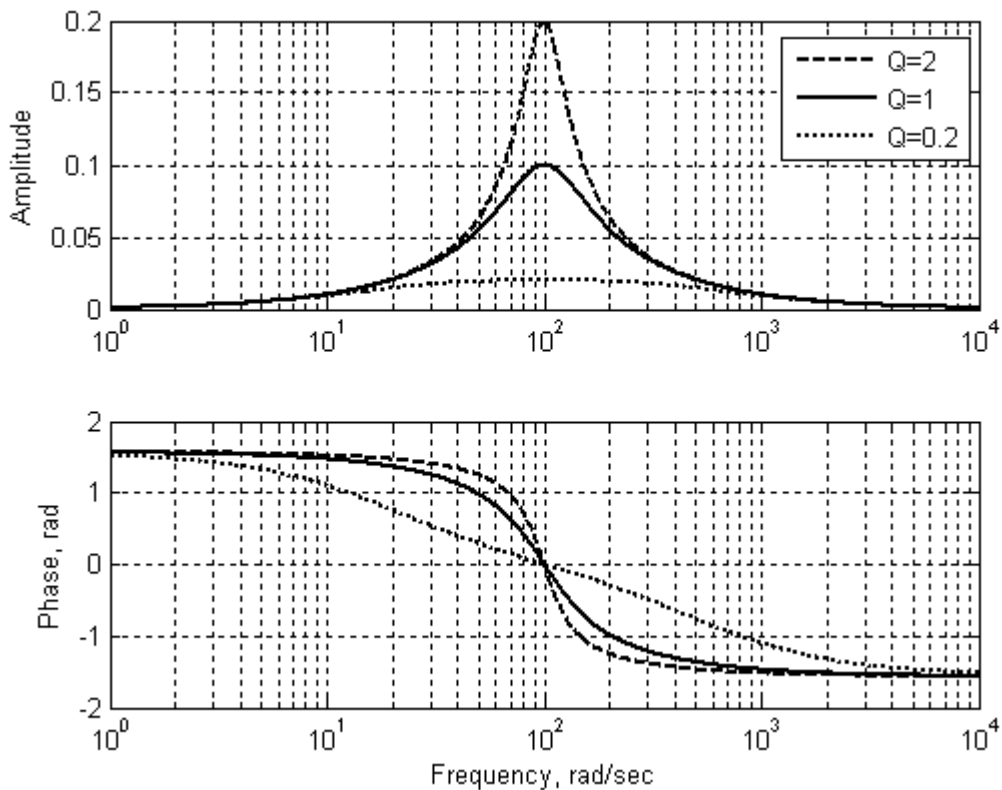


Рис. 166. АЧХ и ФЧХ активного полосопропускающего фильтра

В отличие от характеристик рассмотренных выше фильтров низкой и высокой частоты данный фильтр за пределами полосы пропускания ослабляет сигналы как низких, так и высоких частот. При $Q \ll 1$ имеем фильтр с коэффициентом передачи, практически равным единице в широкой полосе ($f_n \ll f_0$). При $Q \gg 1$ частотная характеристика фильтра имеет четко выраженный резонансный характер вблизи частоты ω_0 .

Чтобы определить переходную характеристику полосопропускающего активного фильтра, вычислим интеграл

$$h(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{K_1}{(s-s_1)(s-s_2)} e^{st} ds$$

Разложив стоящую под интегралом дробь на простейшие,

$$\frac{1}{(s-s_1)(s-s_2)} = \frac{1}{s_1-s_2} \frac{1}{s-s_1} + \frac{1}{s_2-s_1} \frac{1}{s-s_2}$$

на основании (42) получим

$$h(t) = \frac{K_1}{s_1 - s_2} e^{s_1 t} + \frac{K_1}{s_2 - s_1} e^{s_2 t}$$

Характер зависимости $h(t)$ при различных Q представлен на (Рис. 167).

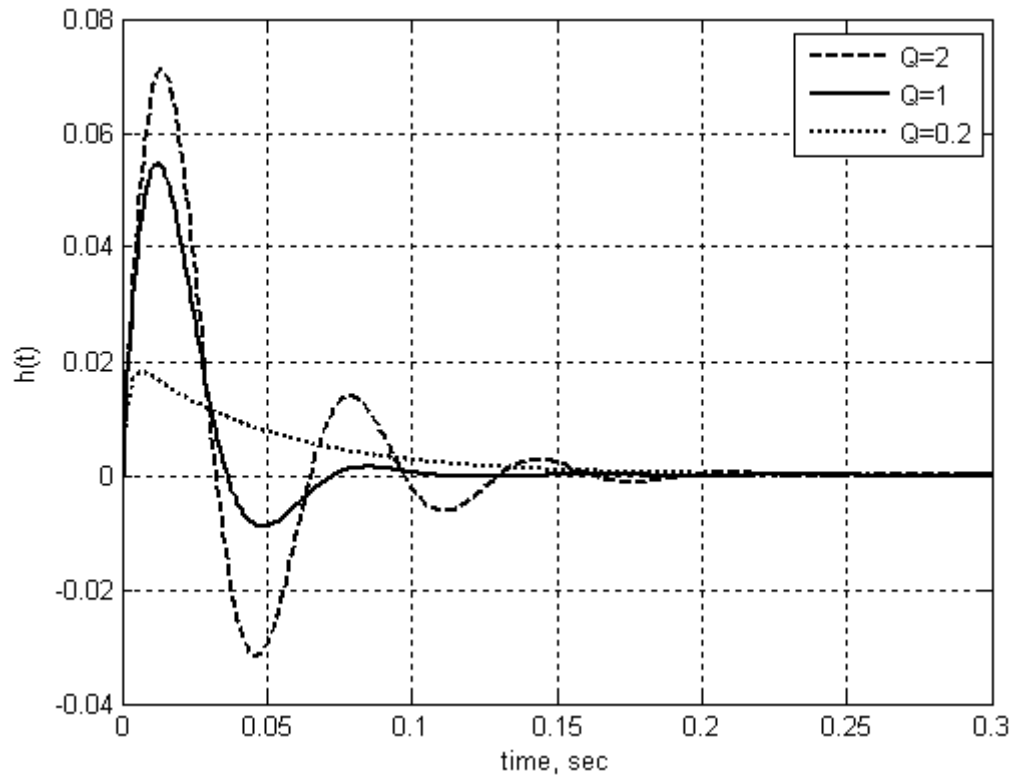


Рис. 167. Зависимость $h(t)$ при различных Q для активного полосопропускающего фильтра

Программа в **MATLAB**, рассчитывающая и строящая графики $h(t)$ для различных Q (с помощью троеточия в **MATLAB** производится перенос длинного выражения на следующую строку):

```
t=0:0.001:0.3;
K1=10;
w0=100;
Q=2;
h1=K1/w0/sqrt(1-1/4/Q^2).* ...
    exp(-w0*t/2/Q).*sin(w0*sqrt(1-1/4/Q^2)*t);
Q=1;
h2=K1/w0/sqrt(1-1/4/Q^2).* ...
    exp(-w0*t/2/Q).*sin(w0*sqrt(1-1/4/Q^2)*t);
Q=0.2;
h3=K1/w0/sqrt(1-1/4/Q^2).* ...
    exp(-w0*t/2/Q).*sin(w0*sqrt(1-1/4/Q^2)*t);
```

```

set (figure, 'Color', 'w')
plot(t, h1, '--k', t, h2, 'k-', t, h3, 'k:', 'LineWidth', 2)
legend('Q=2', 'Q=1', 'Q=0.2'), grid

```

Приведенные выше примеры схем активных фильтров показывают, что использование одного каскада усилителя с частотнозависимой обратной связью, включающей только RC -элементы, позволяет реализовать все возможные характеристики фильтров второго порядка без использования индуктивностей. Более сложные фильтры могут быть созданы каскадным включением рассмотренных выше типовых схем. Отметим, что такой способ синтеза сложных фильтров не является единственным.

Приведем примеры моделирования рассмотренных выше фильтров с помощью **SimPowerSystems MATLAB**. Модельная схема для примера №1 показана на Рис. 168 (в приложении модель **filter2_band_1.mdl**). Для получения переходной характеристики пропустим через систему сигнал «единичного включения» (блок **step**), а АЧХ и ФЧХ вычислим из переходной характеристики, уже рассмотренным раньше способом (дифференцируя и беря быстрое преобразование Фурье).

Так как схема может быть реализована не с любым набором параметров Q , ω_0 , K_I , выберем для простоты расчета следующие параметры схемы:

$$Q = 2, \omega_0 = 100 \text{ рад/с}, C_1 = C_2 = 10^{-6} \text{ Ф.}$$

По этим параметрам из формул (53) можно вычислить $R_1 = 25 \cdot 10^2 \text{ Ом}$, $R_2 = 4 \cdot 10^4 \text{ Ом}$ и $K_I = 4 \cdot 10^2$. Максимальное значение АЧХ для такого фильтра должно быть больше в 40 раз по сравнению с максимальным значением характеристики, показанной на Рис. 166, для $K_I = 10$.

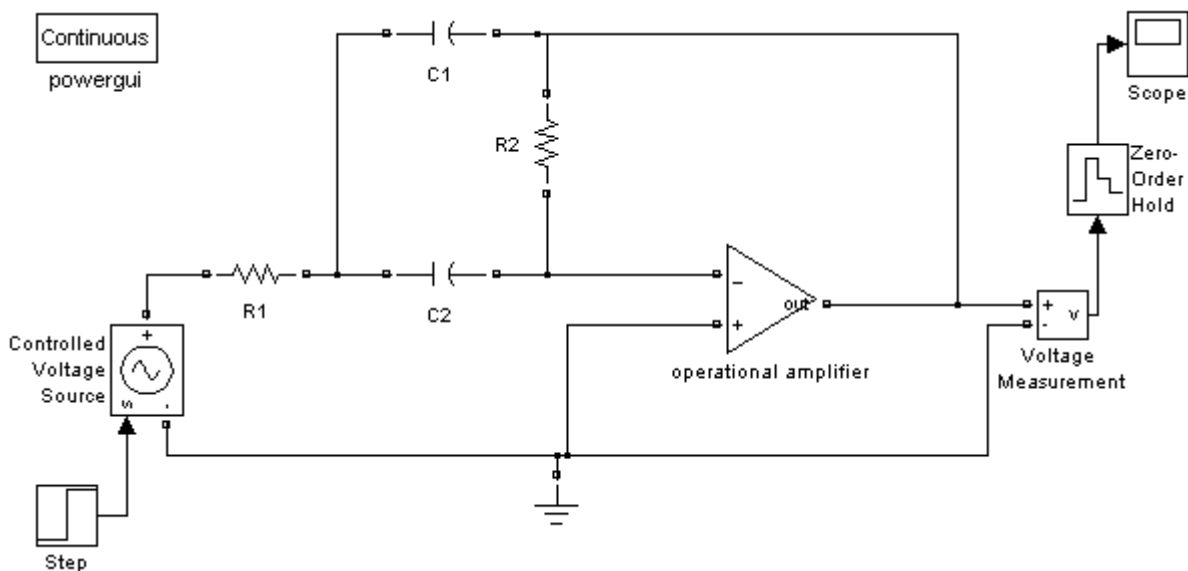


Рис. 168. Схема в SPS для примера №1

Переходная характеристика для этой схемы с выбранными параметрами изображена на Рис. 169.

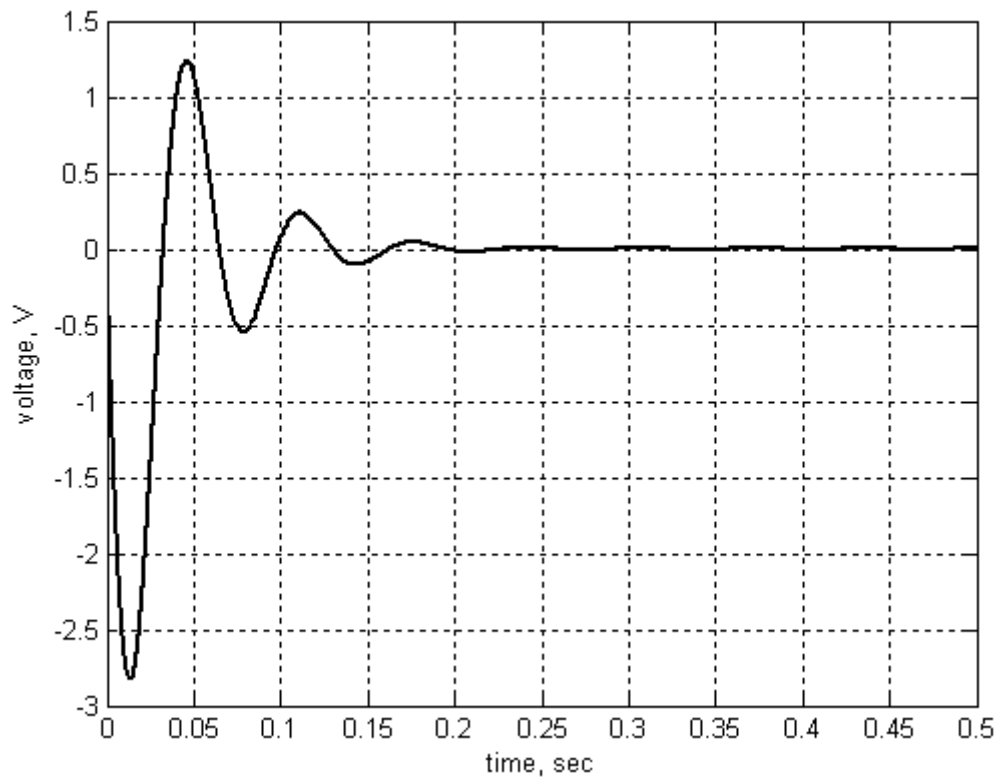


Рис. 169. Переходная характеристика полосового фильтра

Амплитудно- и фазочастотные характеристики изображены на Рис. 170.

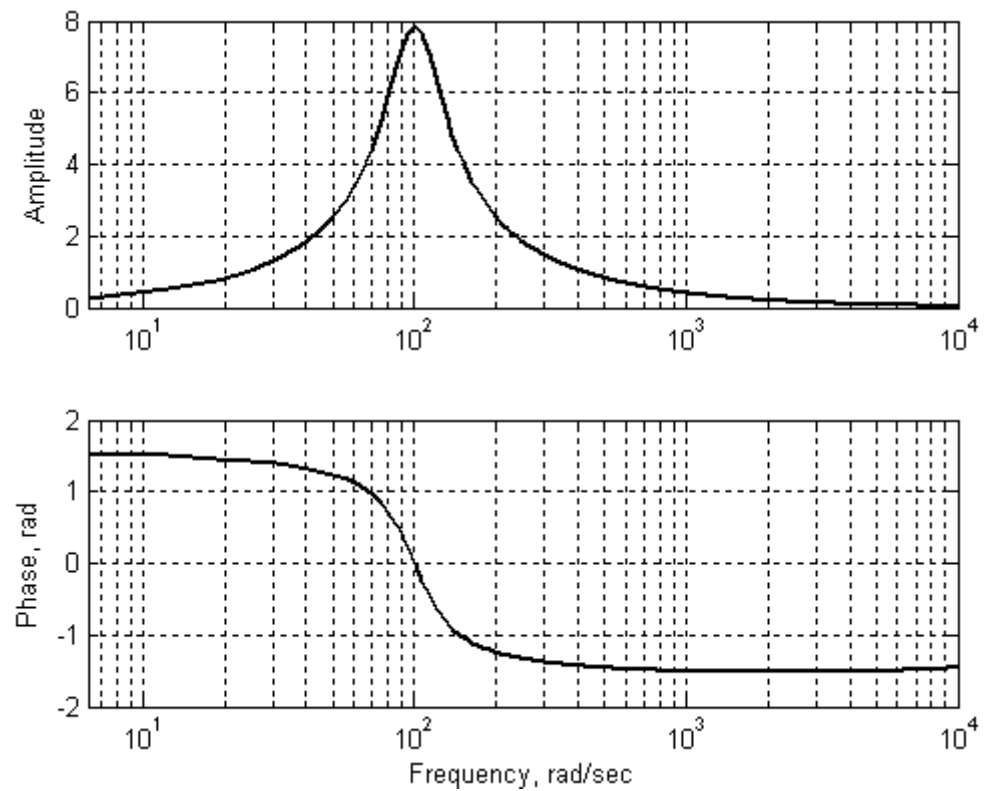


Рис. 170. АЧХ и ФЧХ полосового фильтра

Здесь, в отличие от ранее рассмотренных примеров, нужно было провести дополнительные операции для получения ФЧХ. Чтобы получить фазовую характеристику в обычном виде (согласно Рис. 166), с помощью функции **unwrap** удаляются скачки на 2π . В результате фазовая характеристика приобретает гладкий вид (в диапазоне от 0 до 2π) и отрицательный наклон. Потом значения фазы смещаются на π (изменение в диапазоне от $-\pi$ до π). Заметим, что частота квантования, установленная блоком **Zero-Order Hold**, должна совпадать с частотой f_s . Ниже приведен листинг программы в **MATLAB**:

```
fs=2*pi*10000; %частота квантования
y=ScopeData.signals.values;
g=diff(y)*fs;
N=length(g);
K=fft(g)/fs; %передаточная функция
w=2*pi*(0:N-1)*fs/N; %создаем вектор частот
set(figure,'Color','w') %белый цвет окна
subplot(2,1,1),
semilogx(w,abs(K),'k','LineWidth',2),grid %АЧХ
phase=unwrap(angle(K))-pi; %фаза
subplot(2,1,2),
semilogx(w,phase,'k','LineWidth',2),grid %ФЧХ
```

Можно объединить **Simulink**-модель и приведенную выше программу в **MATLAB**, чтобы после моделирования в **Simulink** автоматически построились графики АЧХ и ФЧХ. Для этого можно использовать **Callback** функции **Simulink**-модели. В меню рабочего окна **Simulink**-модели выбираем **File/Model properties**. В появившемся окне **Model properties** выбираем вкладку **Callbacks** (Рис. 171). В правой части этого окна перечислены функции (программы), которые будут выполняться при некоторых условиях, например, до или после загрузки модели, после начала моделирования и т. д. Мы будем использовать функцию **StopFcn**, которая выполняется после завершения моделирования. В левой части окна **Callbacks** нужно записать код выполняемой программы. В данном случае (Рис. 171) это чуть измененная программа **MATLAB**, уже разобранный выше в этом примере. Заметим, что в этой программе не задана переменная f_s – частота квантования. Параметр **Sample time** (величина обратная частоте квантования) также используется блоком **Zero-Order Hold**. Для того чтобы частоты квантования не отличались, мы используем параметр $1/f_s$ в настройках блока **Zero-Order Hold** (Рис. 172). А саму величину $f_s = 2\pi 10000$ (для данного примера) задаем как функцию **PostLoadFcn**, которая выполняется после загрузки модели (можно использовать **InitFcn**, которая выполняется при инициализации модели). В этой функции происходит создание единственной переменной (Рис. 172). Теперь, после загрузки модели и выполнения моделирования графики АЧХ и ФЧХ (Рис. 170) будут строиться автоматически.

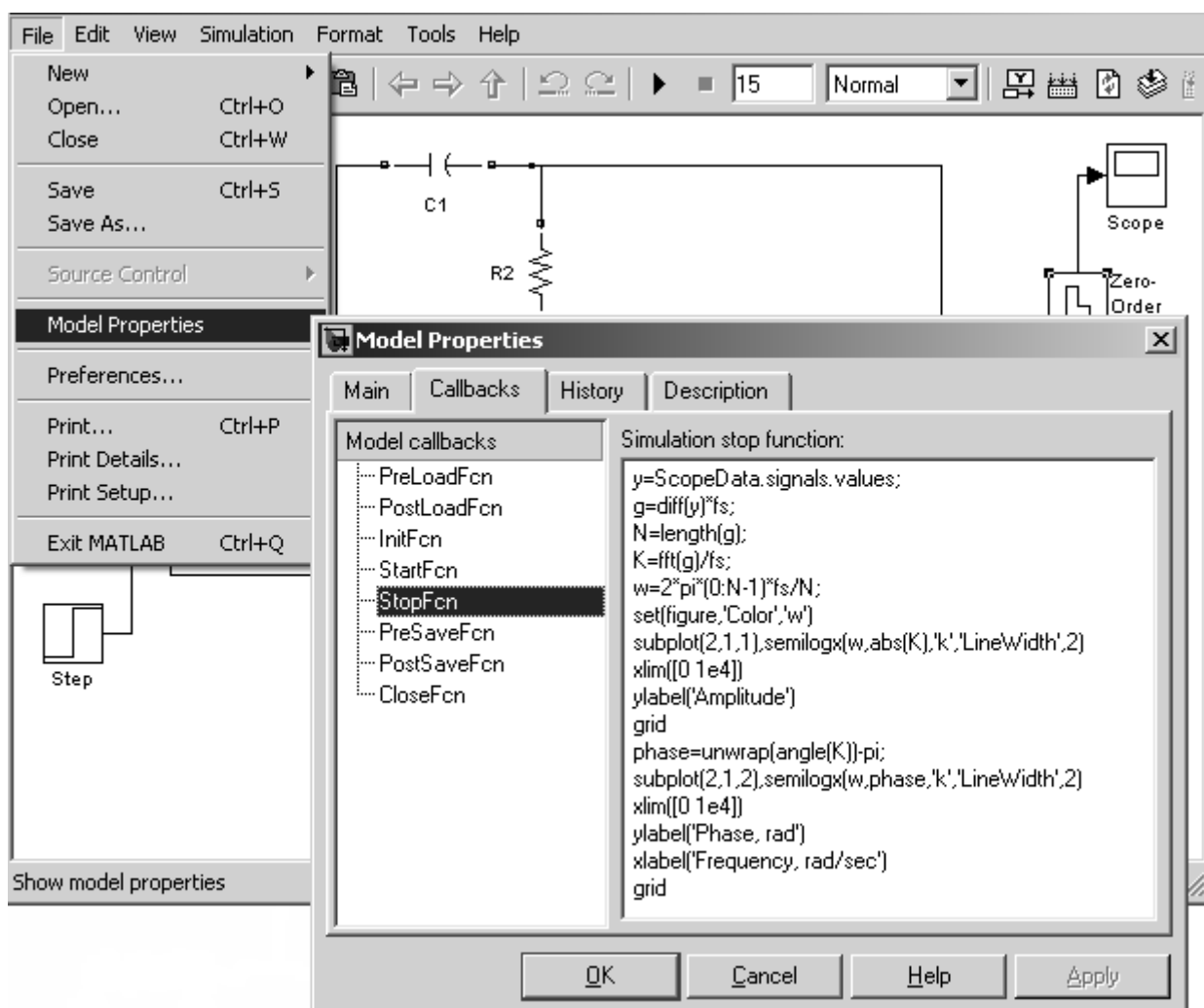


Рис. 171. Использование Callback функции

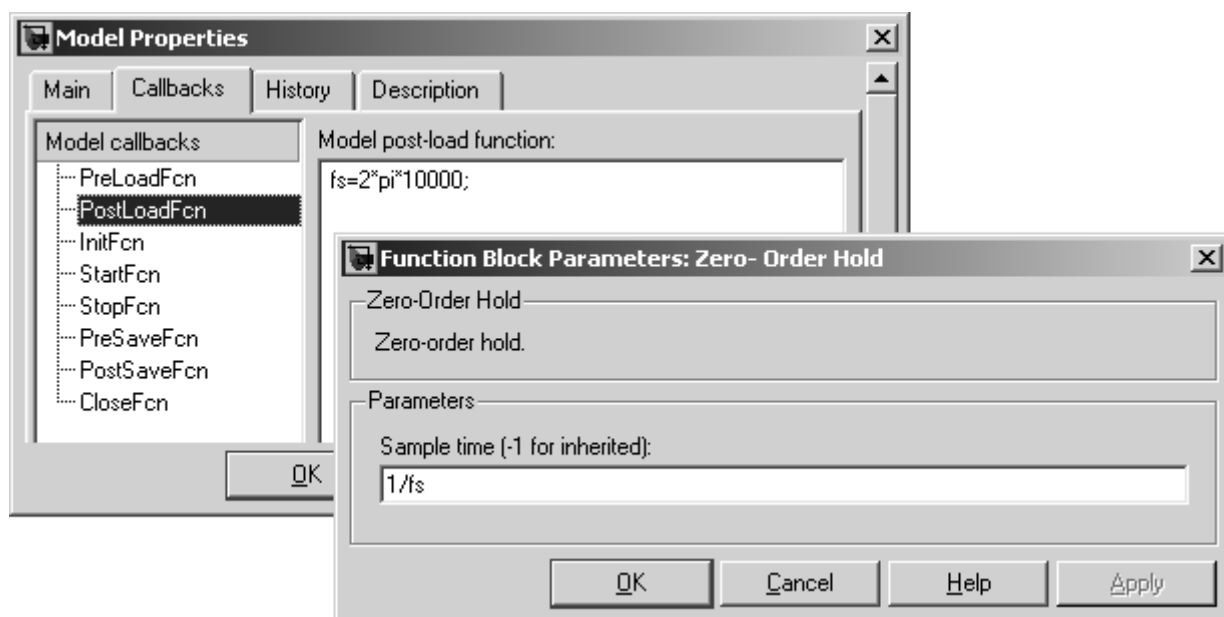


Рис. 172. Задание частоты квантования

Для примера №2 схема в SPS будет выглядеть, как показано на Рис. 173 (в приложении модель **filter2_band_2.mdl**). Для удобства зададим параметры схемы $Q = 2$, $\omega_0 = 100$ рад/с, $R_3 = R_2 = 10^3$ Ом и по формулам (54) рассчитаем остальные параметры: $C_4 = 25 \cdot 10^{-7}$ Ф, $C_1 = 4 \cdot 10^{-5}$ Ф, $K_I = 400$.

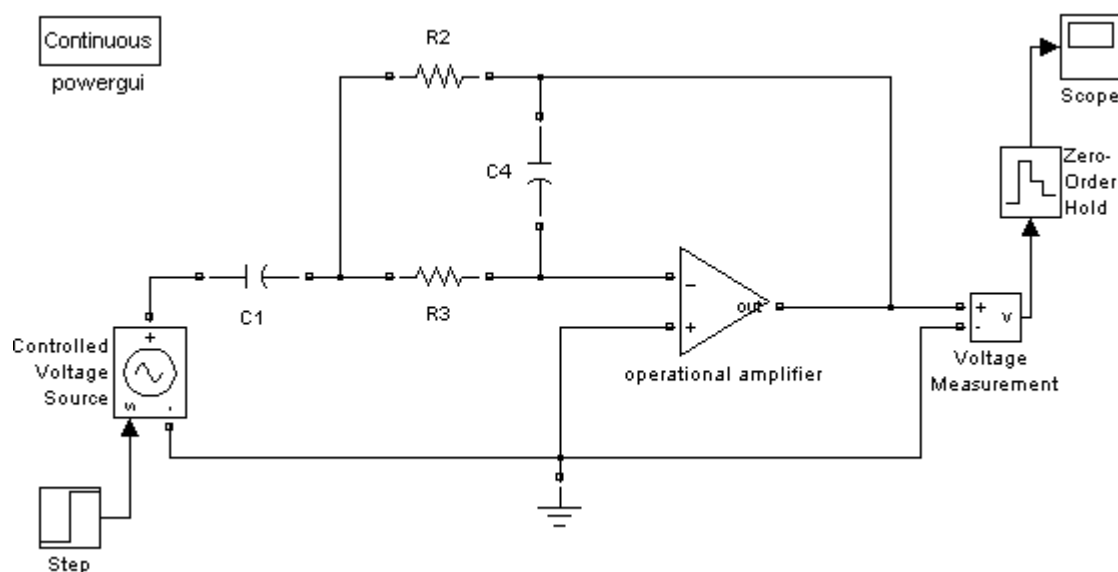


Рис. 173. Схема в SPS для примера №2

Переходная характеристика совпадает с изображенной на Рис. 169. АЧХ и ФЧХ приведены на Рис. 174.

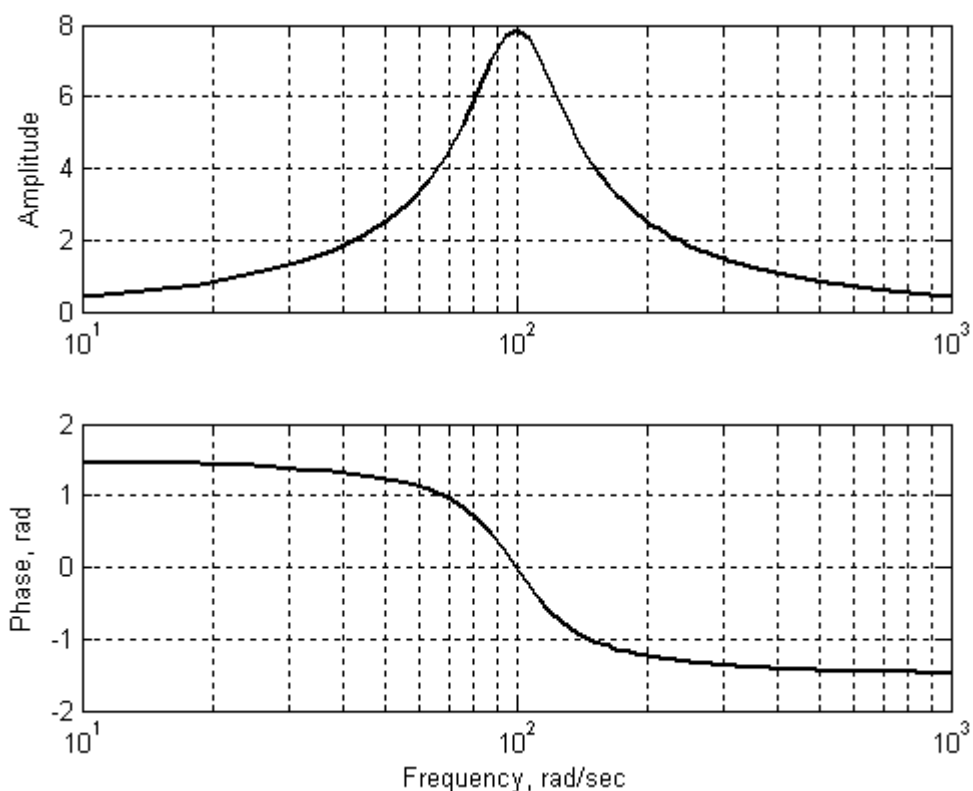


Рис. 174. АЧХ и ФЧХ полосового фильтра

Для сравнения, приведем пример моделирования RLC – фильтра. В приложении этой схеме соответствует модель **filter2_band_RLC.mdl**. Для значений $Q = 2$, $\omega_0 = 10^6$ рад/с рассчитаем параметры схемы из формул (55) $C=10^{-9}$ Ф, $L=10^{-3}$ Гн, $R=5 \cdot 10^2$ Ом. Изображение SPS модели показано на Рис. 175.

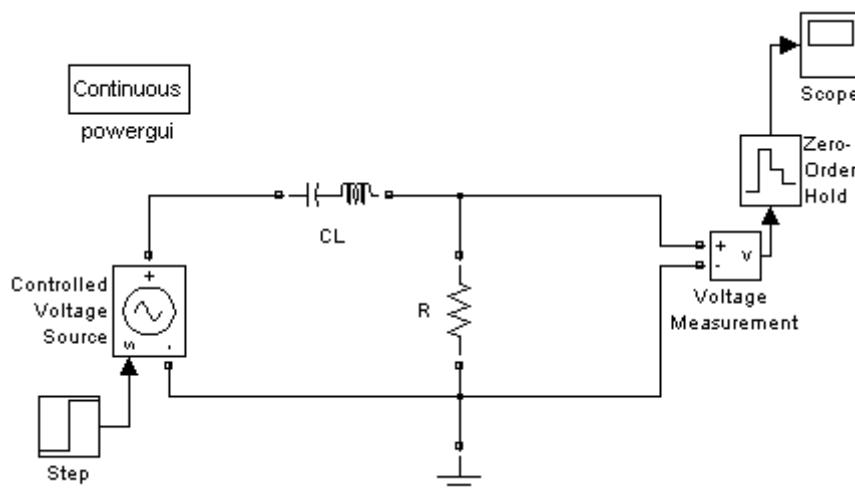


Рис. 175. Схема SPS для RLC – фильтра

Переходная характеристика для выбранных параметров фильтра изображена на Рис. 176.

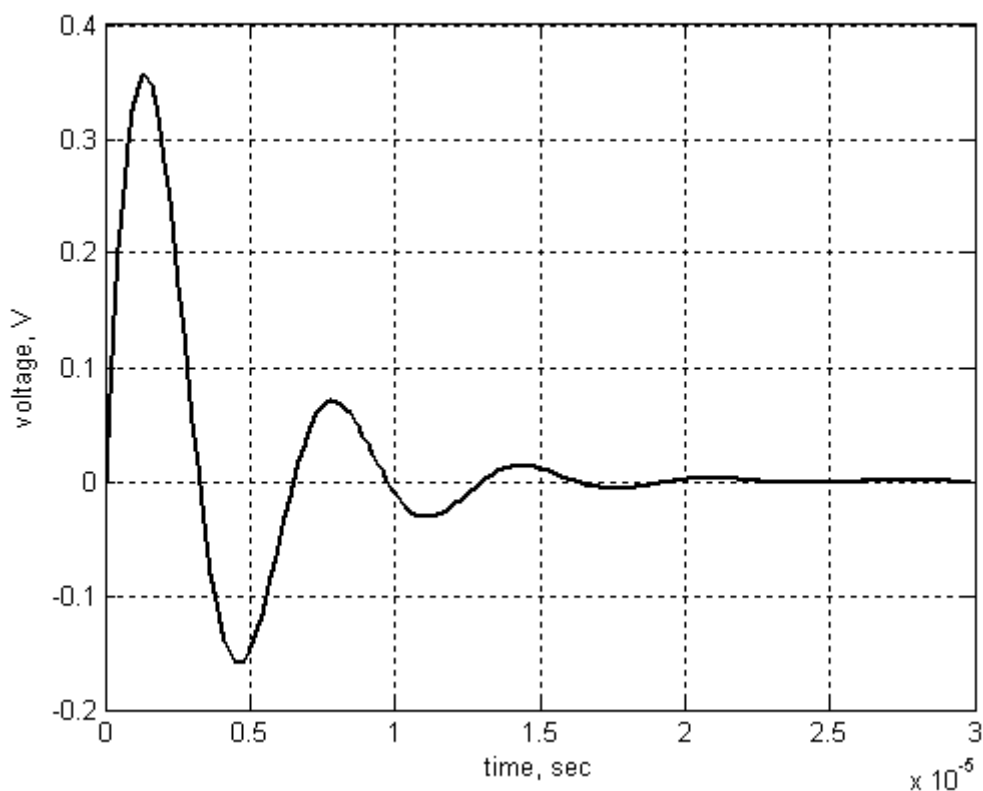


Рис. 176. Переходная характеристика для RLC – фильтра

АЧХ и ФЧХ показаны на Рис. 177.

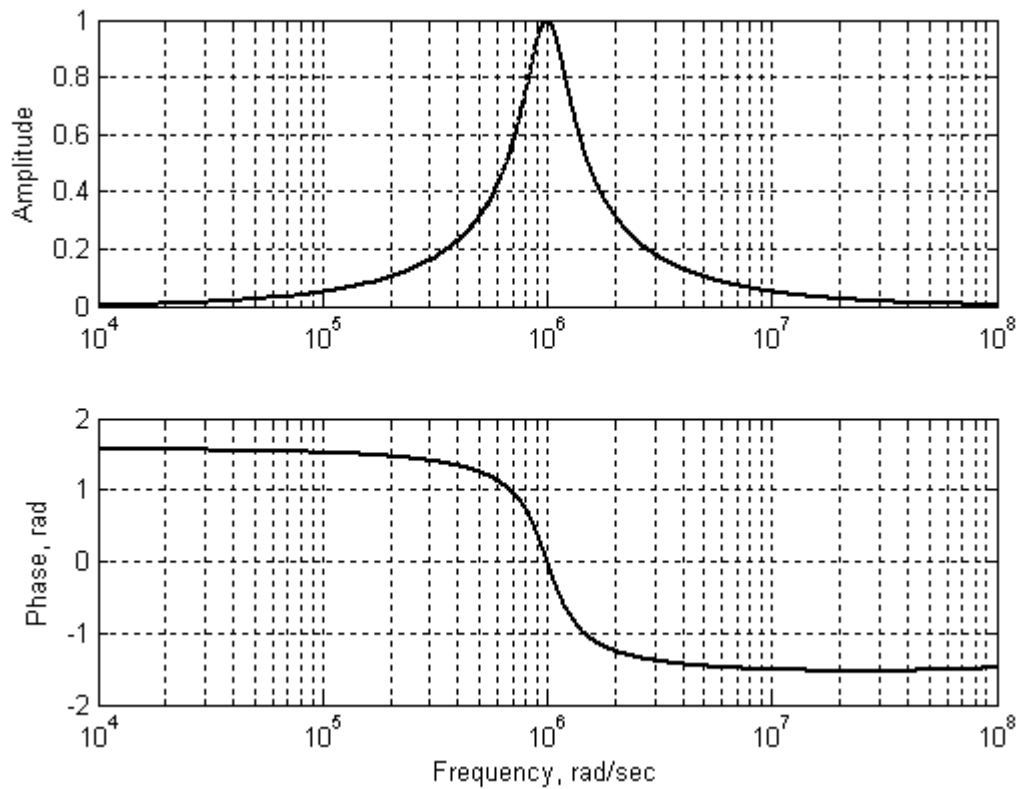


Рис. 177. АЧХ и ФЧХ для RLC – фильтра

5.1.9. Активные фильтры первого порядка

Рассмотрим теперь активные фильтры первого порядка. В качестве исходной возьмем схему второго типа (Рис. 161) с коэффициентом передачи (50). Если положить $Z_3 = 0$, $Z_4 = \infty$, то придем к следующей схеме активного фильтра первого порядка (Рис. 178).

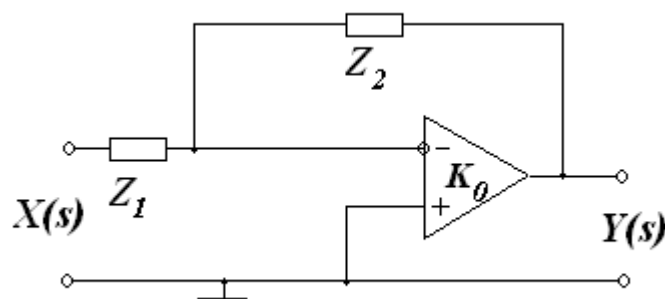


Рис. 178. Активный фильтр первого порядка

Коэффициент передачи такого фильтра будет

$$K(s) = -\frac{K_0 Z_2}{Z_1(K_0 + 1) + Z_2} \quad (56)$$

Выбором Z_1 , Z_2 можно реализовать активные фильтры низких и высоких частот. Активный фильтр высоких частот изображен на Рис. 179, а его пассивный аналог – на Рис. 180.

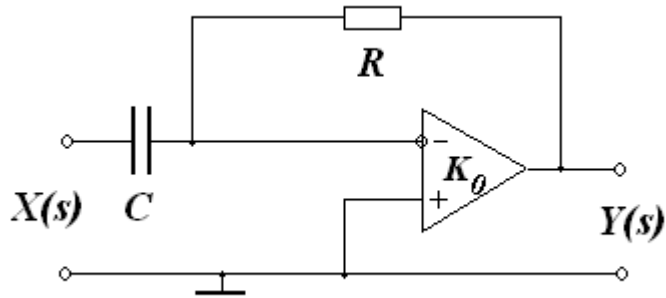


Рис. 179. Активный фильтр первого порядка высоких частот

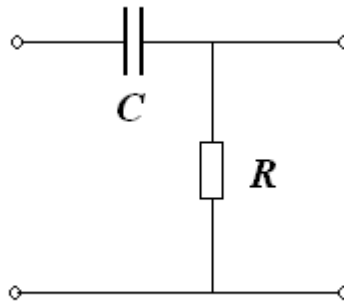


Рис. 180. Пассивный фильтр первого порядка высоких частот

Здесь $Z_1 = 1/sC$ и $Z_2 = R$. Используя (56), получаем

$$K(s) = -\frac{K_0 s \tau_1'}{1 + s \tau_1'}, \quad \tau_1' = \frac{RC}{K_0 + 1}. \quad (57)$$

Для коэффициента передачи пассивного фильтра имели

$$K(s) = \frac{s \tau}{1 + s \tau}, \quad \tau = RC.$$

Сравнивая приведенные выражения для коэффициентов передачи видим, что при одинаковых постоянных τ_1' и τ модуль коэффициента передачи активного фильтра в K_0 раз больше, чем у пассивного.

Частотные характеристики пассивного фильтра высоких частот приведены на Рис. 181 (для $\tau = 1$ с).

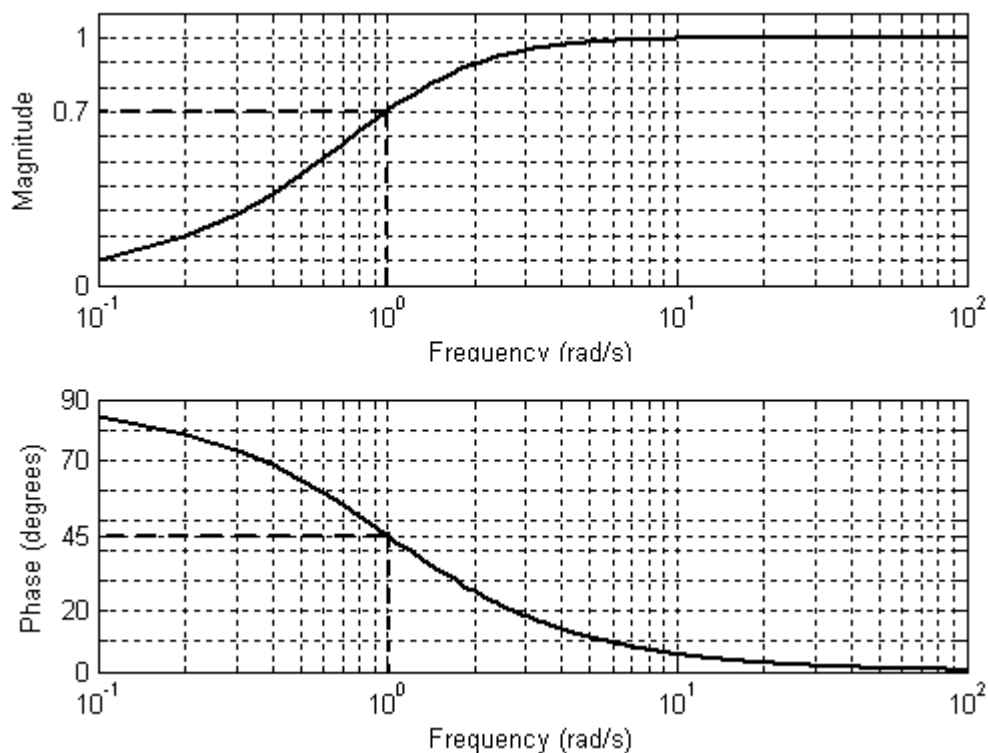


Рис. 181. АЧХ и ФЧХ пассивного фильтра высоких частот (для $\tau = 1\text{с}$)

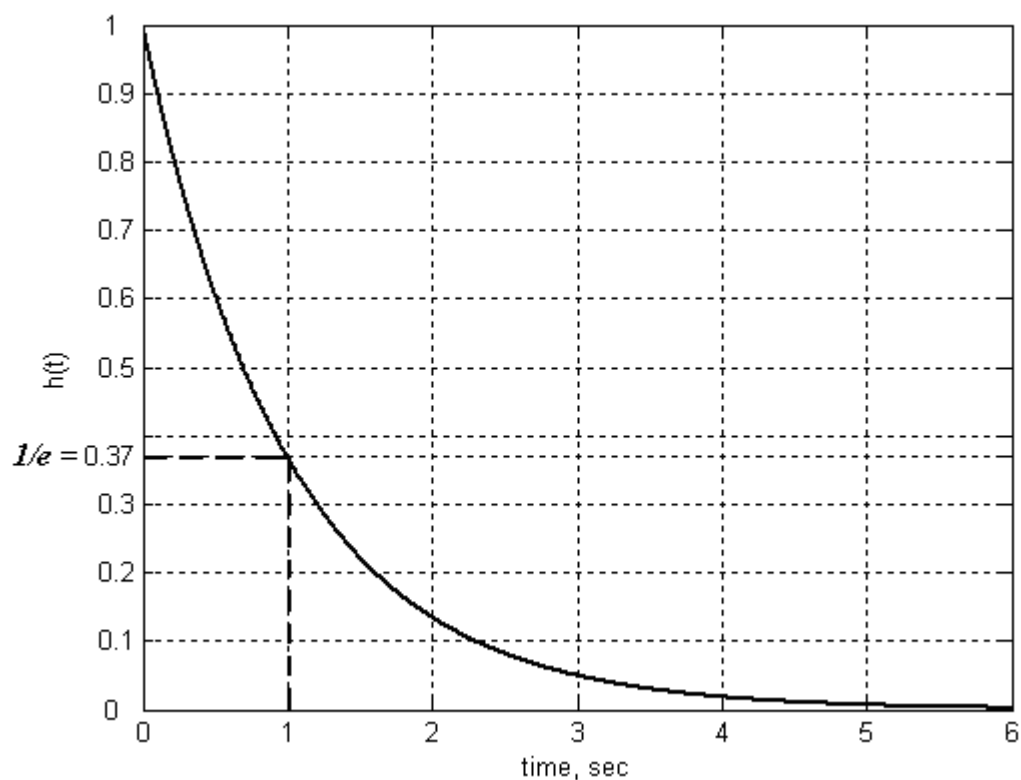


Рис. 182. Переходная характеристика пассивного фильтра высоких частот

Переходная характеристика активного фильтра высоких частот

$$h(t) = -\frac{K_0 \tau_1'}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{1+s\tau_1'} e^{st} ds = -K_0 e^{-\frac{t}{\tau_1'}}$$

также с точностью до знака («минус» появляется из-за того, что сигнал на выходе подается на инвертирующий вход ОУ) совпадает с переходной характеристикой пассивного фильтра, график которой изображен на Рис. 182.

Частотные и временные характеристики для модели пассивного фильтра в **SPS** можно получить с помощью инструмента **LTI-viewer**, доступ к которому находится в блоке **Powergui**. Однако, для модели активного фильтра высоких частот в **SPS**, изображенной на Рис. 183, **LTI-viewer** дает неверные результаты.

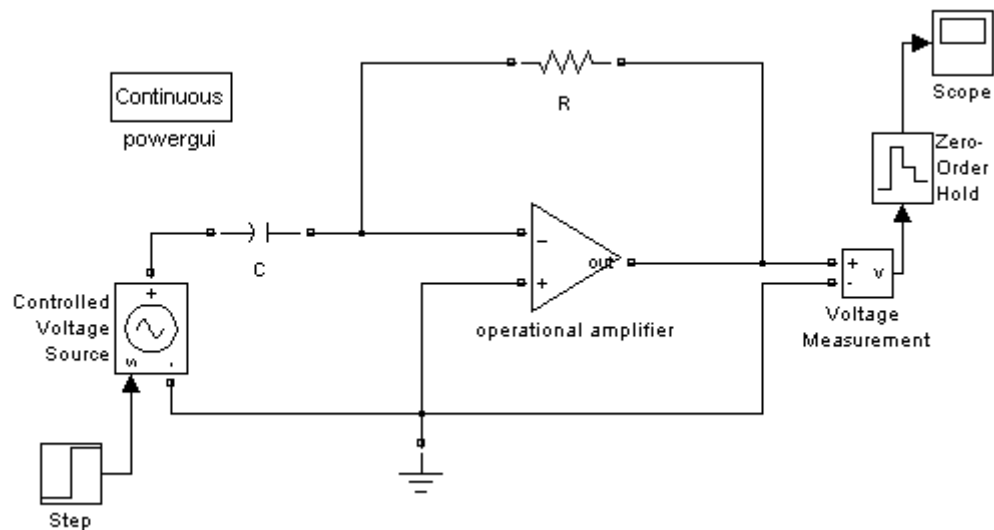


Рис. 183. Модель активного фильтра высоких частот

Поскольку мы рассматриваем характеристики для схемы с идеальным ОУ, необходимо оставаться в линейном диапазоне напряжений для ОУ. Т. к. коэффициент усиления по постоянному напряжению K_0 очень велик, то ОУ быстро входит в режим насыщения. На практике этого можно избежать, уменьшая напряжение входного сигнала «ступеньки». Для моделирования достаточно просто убрать из подсистемы ОУ блок насыщения (Рис. 184).

Для коэффициента усиления ОУ $K_0 = 1000$ выберем параметры схемы $C = 10^{-3}$ Ф и $R = 10^6$ Ом, чтобы $\tau_1' = 1$ с в формуле 57. После моделирования переходная характеристика активного фильтра высоких частот для этих параметров изображена на Рис. 185.

До включения «ступеньки» ($t < 0$) сигнал на выходе фильтра равен нулю. Как мы получили (Рис. 185), в момент времени $t = 0$ переходная характеристика скачком принимает значение $-K_0$. Заметим что в случае, когда переходная характеристика имеет скачок, уже нельзя будет использовать метод получения АЧХ и ФЧХ с помощью дифференцирования переходной характеристики.

Покажем, почему этот метод не работает на примере пассивного фильтра (переходная характеристика которого также имеет скачок (Рис. 182))

и далее рассмотрим еще один способ нахождения частотных характеристик и его реализацию в **MATLAB**.

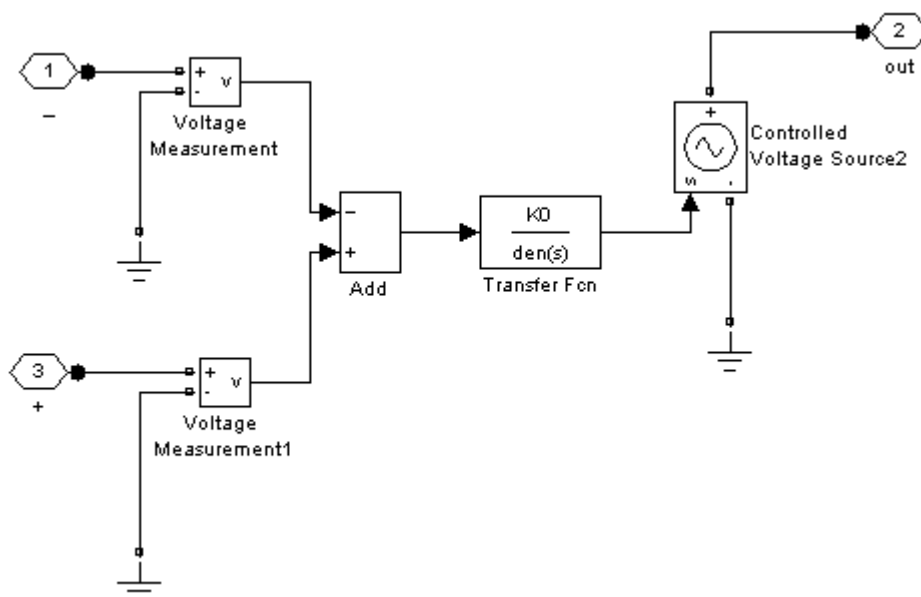


Рис. 184. Подсистема операционного усилителя

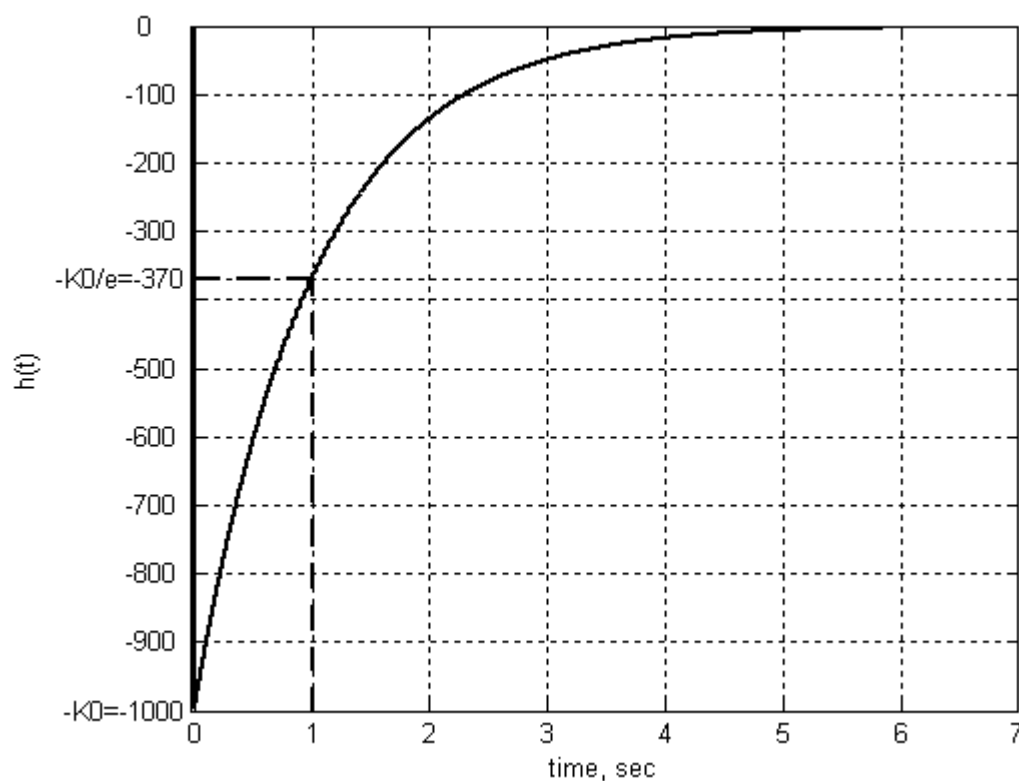


Рис. 185. Переходная характеристика активного фильтра высоких частот (для $\tau_I' = 1$ с)

Для пассивного фильтра высоких частот (дифференцирующей цепочки), изображенной на Рис. 180, можно написать:

$$dU_{\text{вых}} = dU_C + dU_R = dU_C + dU_{\text{вх}}.$$

Выразив в этом выражении изменение напряжения на конденсаторе dU_C как

$$dU_C = \frac{dq}{C} = \frac{I_C dt}{C}$$

и учитывая, что ток, проходящий через конденсатор и реостат, одинаков $I_C = I_R = \frac{U_R}{R} = \frac{U_{\text{вх}}}{R}$, получаем

$$dU_{\text{вых}} = \frac{U_{\text{вх}} dt}{RC} + dU_{\text{вх}}.$$

Отсюда можно получить дифференциальное уравнение для дифференцирующей цепочки:

$$\frac{dU_{\text{вых}}}{dt} = \frac{U_{\text{вх}}}{RC} + \frac{dU_{\text{вх}}}{dt}.$$

Введем новые обозначения: напряжение на входе будем обозначать буквой x , а напряжение на выходе – буквой y и введем постоянную времени цепочки τ . Тогда это уравнение можно переписать как:

$$\frac{dx}{dt} = \frac{1}{\tau} y + \frac{dy}{dt} \quad (58)$$

Ищем его переходную характеристику $h(t)$. Для этого в левую часть уравнения (на вход) подставляем функцию Хевисайда (единичную ступеньку):

$$x(t) = \theta(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

Как известно, производная от этой функции является дельта-функцией $\delta(t)$:

$$\frac{d\theta}{dt} = \delta(t)$$

Получаем уравнение (58):

$$\delta(t) = \frac{1}{\tau} h + \frac{dh}{dt}$$

Решаем это уравнение спектральным методом.

$$\delta(t) \rightarrow 1 \text{ и } \frac{d}{dt} \rightarrow i\omega,$$

Переходная характеристика находится, как обратное преобразование Фурье. Интеграл вычисляем по вычетам методом контурного интегрирования. Полюс подынтегрального выражения $\frac{i}{\tau}$. Если $t > 0$, подынтегральное выражение при $\omega \rightarrow \infty$, затухает в верхней полуплоскости, соответственно там замыкаем контур. При $t < 0$ наоборот, в нижней полуплоскости полюсов нет, поэтому интеграл равен нулю. Результат можно записать с помощью функции единичного скачка $\theta(t)$:

$$h(t) = \frac{1}{2\pi i} \int_{-\infty}^{+\infty} \frac{1}{\omega - \frac{i}{\tau}} e^{i\omega t} d\omega = \begin{cases} e^{-\frac{t}{\tau}}, & t > 0 \\ 0, & t < 0 \end{cases} = \theta(t) e^{-\frac{t}{\tau}}$$

Теперь, зная переходную характеристику $h(t) = \theta(t) e^{-\frac{t}{\tau}}$ дифференцирующей цепочки, находим её коэффициент передачи $K(\omega)$. Коэффициент передачи – это Фурье образ функции Грина $G(t)$, а функция Грина – это производная переходной характеристики.

$$K(\omega) = \int_{-\infty}^{+\infty} G(t) e^{-i\omega t} dt \quad G(t) = \frac{d}{dt} h(t)$$

Переходную характеристику надо дифференцировать по правилам дифференцирования обобщённых функций, так как она имеет разрыв первого рода (скачок). Это значит, что её необходимо дифференцировать как произведение двух функций – экспоненты и $\theta(t)$.

$$G(t) = \frac{d}{dt} h(t) = \delta(t) e^{-\frac{t}{\tau}} - \frac{1}{\tau} \theta(t) e^{-\frac{t}{\tau}}$$

Производная $\theta(t)$ – это $\delta(t)$, а произведение $\delta(t)$ и любой функции – это то же, что и $\delta(t)$, умноженная на значение этой функции в нуле. В результате функция Грина:

$$G(t) = \delta(t) - \frac{1}{\tau} \theta(t) e^{-\frac{t}{\tau}}$$

Чтобы вычислить преобразование Фурье функции Грина пользуемся тем, что $\tau > 0$ и тем, что функция Грина равна нулю при $t < 0$. Пределы интегрирования будут от 0 до бесконечности, и на верхнем пределе экспонента стремится к нулю по модулю. А также преобразование Фурье $\delta(t) \rightarrow 1$.

$$K(\omega) = \int_{-\infty}^{+\infty} G(t) e^{-i\omega t} dt = 1 - \frac{1}{\tau} \int_0^{+\infty} e^{-\left(i\omega + \frac{1}{\tau}\right)t} dt = 1 - \frac{1}{i\omega\tau + 1} = \frac{i\omega\tau}{i\omega\tau + 1}$$

В результате получили коэффициент передачи дифференцирующей цепочки.

Когда эти же действия выполняются в **MATLAB**, не учитываются детали, связанные с тем, что переходная характеристика имеет разрыв первого рода. В результате в функции Грина теряется $\delta(t)$, и при выполнении преобразования Фурье функции Грина, пропадает 1. Потеря этой единицы приводит к тому, что в результате получается коэффициент передачи интегрирующей цепочки.

Чтобы устранить эту проблему, можно подать на вход цепочки сигнал с достаточно широким спектром, а потом восстановить коэффициент передачи, используя быстрое преобразование Фурье. В качестве такого сигнала можно использовать очень короткий импульс, обладающий свойствами дельта-функции (однородным спектром) в определенном диапазоне частот. Рассмотрим схему активного фильтра в **SPS** (Рис. 186). В приложении эта модель называется **filter2_up.mdl**.

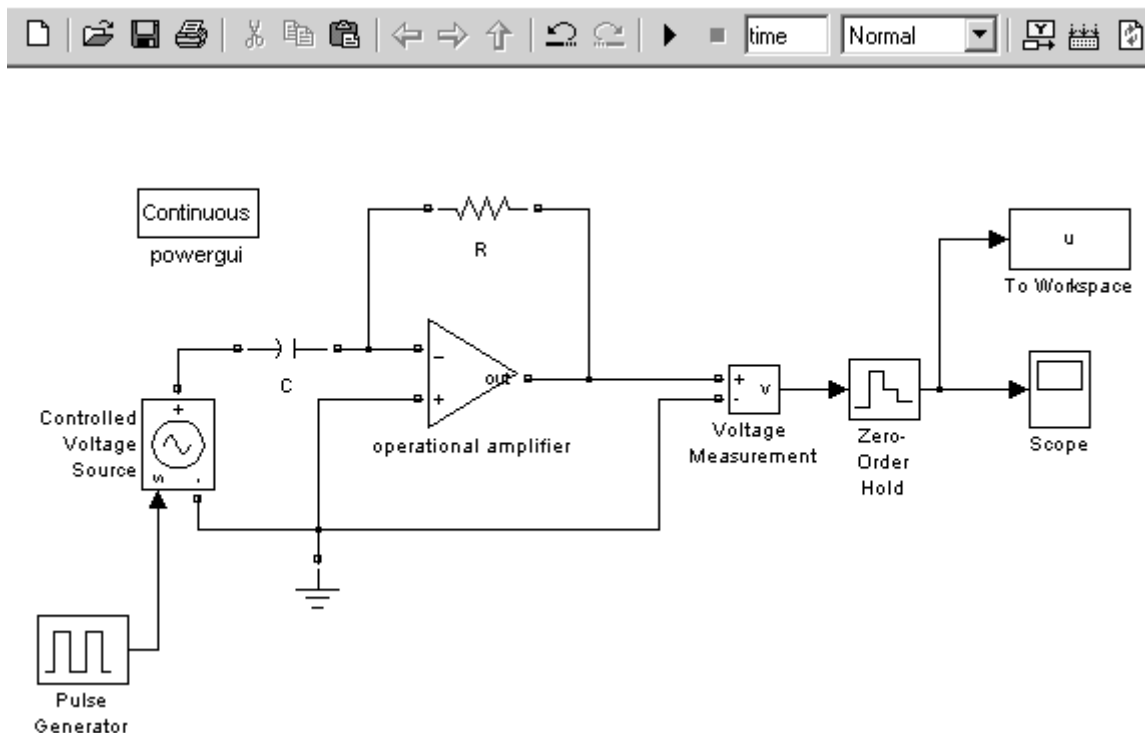


Рис. 186. Получение АЧХ и ФЧХ для активного фильтра высоких частот

На вход фильтра, который ограничен блоками **Controlled Voltage Source** и **Voltage Measurement**, подаём прямоугольный импульс длительностью τ_u и амплитудой $1/\tau_u$. Длительность импульса в блоке **Pulse Generator** выбираем такой, чтобы за время моделирования (обозначим переменной *time*) через систему прошел только один импульс. Этот импульс играет роль дельта-функции. На Рис. 187 изображены амплитудный спектр прямоугольного импульса $|X(\omega)|$, который имеет «лепестковую» структуру, и АЧХ исследуемого фильтра $|K(\omega)|$ в интересующем нас диапазоне частот $0 < \omega < \omega_{max}$. Из рисунка видно, что чтобы получить АЧХ и ФЧХ без искажений, ко-

торые вызваны изменением $|X(\omega)|$ с частотой, необходимо, чтобы $\omega_{max} \ll 2\pi/\tau_u$, то есть интересующий диапазон находился целиком внутри первого лепестка амплитудного спектра прямоугольного импульса.

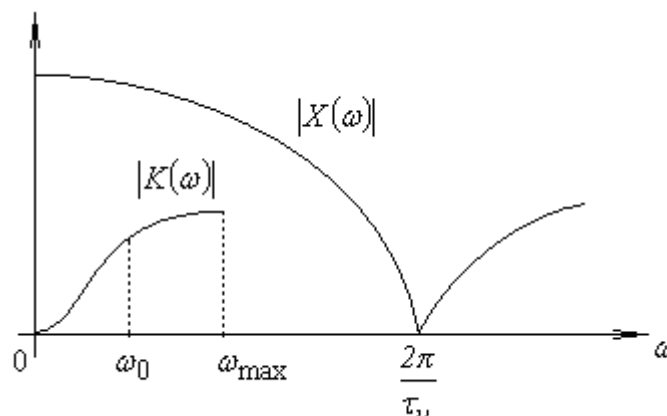


Рис. 187. Пояснение к условию выбора длительности импульса

Если задана ω_{max} , то длительность импульса можно выбрать $\tau_u = \frac{2\pi}{100\omega_{max}}$. На Рис. 188 показаны параметры блока **Pulse Generator**. В переменной *tau* хранится значение величины τ_u . Скважность сигнала выбираем такой, чтобы длительность импульса была τ_u .

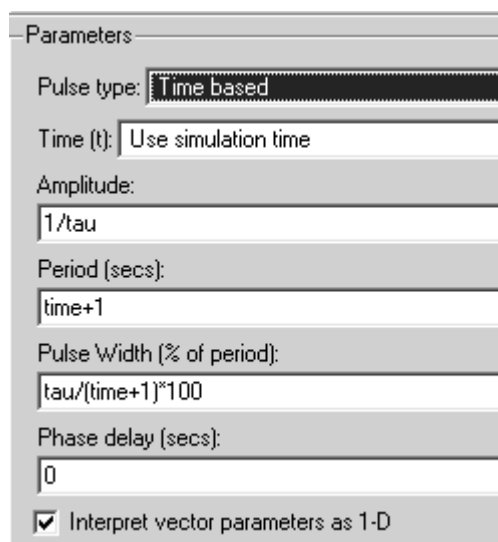


Рис. 188. Параметры блока Pulse Generator

Частоту квантования необходимо выбирать, исходя из теоремы Котельникова. Выражение $f_s = \frac{10}{\tau_u}$ удовлетворяет этому условию. Частота квантования, как и прежде, используется в блоке **Zero-Order Hold**.

Пусть мы хотим построить график АЧХ в интервале $0 < \omega < \omega_{\max}$, используя N точек. Задав N , определяем необходимую для этого длительность записи, то есть конечное время моделирования $T = \frac{2\pi N}{\omega_{\max}}$.

Задача ставится таким образом, что входными параметрами должны быть ω_{\max} и N . Остальные параметры: τ_u , f_s и T рассчитываются из них. В модели величины ω_{\max} , N , τ_u , f_s и T обозначены переменными `omega`, `N`, `tau`, `fs` и `time` соответственно. Эти переменные задаются в **Callback** функции **PostLoadFcn**:

```
omega=1e2;  
N=1000;  
tau=2*pi/100/omega;  
time=2*pi*N/omega;  
fs=10/tau;
```

Полученный в результате моделирования выходной сигнал после оцифровки выводим на осциллограф и сохраняем в переменной u в рабочей области **MATLAB** с помощью блока **To Workspace**. В окне параметров этого блока имя переменной **Variable name** выбрано как u и формат сохраняемых данных **Save format** указан как **Array**.

Чтобы получить частотные характеристики, необходимо выполнить преобразование Фурье выходного сигнала и результат преобразования Фурье разделить на частоту квантования. Частотной характеристике соответствует только первые N элементов вектора, полученного в результате преобразования Фурье, остальная часть искажена из-за неоднородности спектра прямоугольного импульса, как уже обсуждалось выше. Частота отсечки активного фильтра высоких частот была выбрана как $\omega_0 = 1 \frac{\text{рад}}{c}$, для этого коэффициент усиления $K_0 = 1000$, $R = 1 \text{ МОм}$, $C = 0,001 \text{ Ф}$. Поэтому мы выбираем диапазон частот от 0 до $\omega_{\max} = 100 \frac{\text{рад}}{c}$. Для построения графиков будем использовать $N = 1000$ точек. Тогда, по приведенным выше формулам, находим $\tau_u = 2 \cdot 10^{-4} \pi$ секунд, $f_s = \frac{1}{2\pi} \cdot 10^5$ Гц, время моделирования $time = 20\pi$ секунд. Переменная $time$ задается прямо в окне моделирования Рис. 186.

Для обработки результатов моделирования **Simulink** и построения графиков для модели задана **StopFcn Callback**-функция:

```
K = fft(u)/fs; %коэффициент передачи  
w = (0:N-1)/N*omega; %диапазон частот  
figure('Color','w')  
subplot(2,1,1),  
semilogx(w,abs(K(1:N)),'k','LineWidth',2) %АЧХ  
grid on
```

```

xlabel('\omega, rad/s')
ylabel('Amplitude')
subplot(2,1,2)
semilogx(w,unwrap(angle(K(1:N)))...
          *180/pi-180,'k','LineWidth',2)      %ФЧХ
grid on
xlabel('\omega, rad/s')
ylabel('Phase, deg')

```

В первой строчке результат преобразования Фурье сразу делится на частоту квантования. Во второй строчке определяется вектор со значениями частот. Графики будут строиться по N точкам, поэтому в векторе N элементов, а в командах, строящих графики, используется только $1:N$ элементов вектора K . На Рис. 189 показаны полученные амплитудно-частотные характеристики (для $\tau = 1$ с).

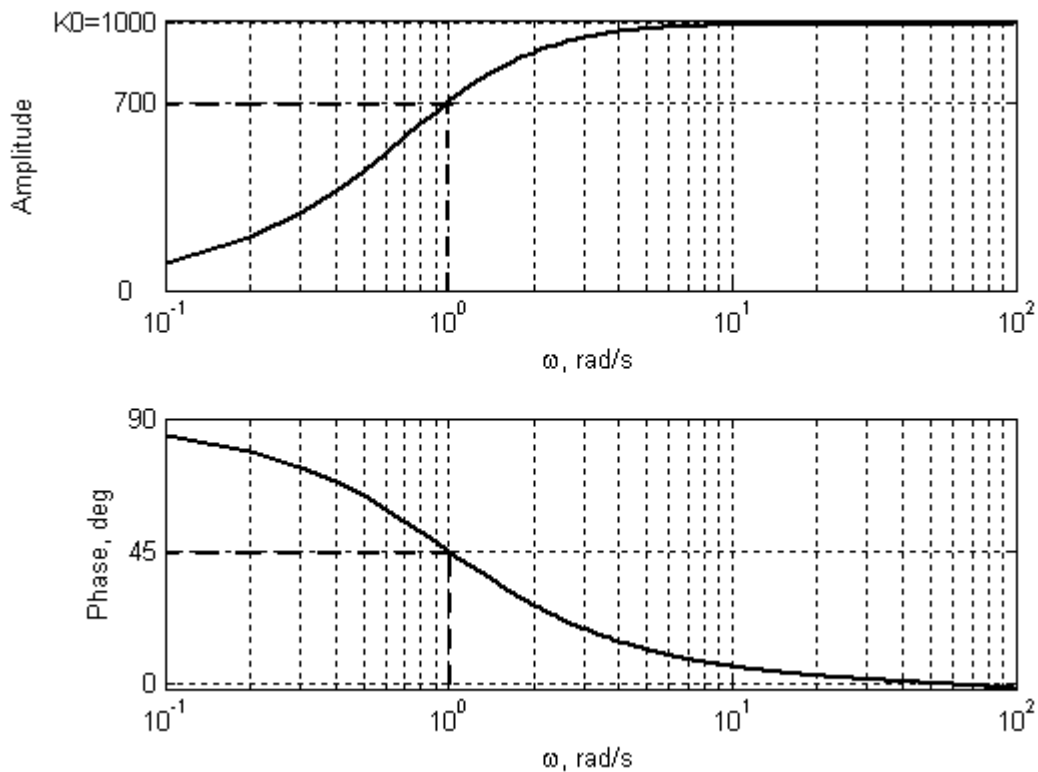


Рис. 189. АЧХ и ФЧХ активного фильтра высоких частот (для $\tau = 1$ с)

Активный фильтр низких частот первого порядка реализуется схемой Рис. 190, где $Z_1 = R$ и $Z_2 = 1/sC$.

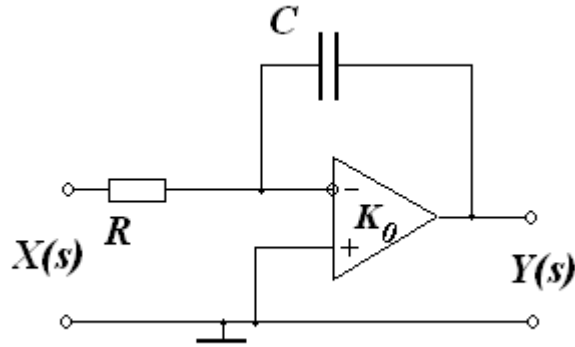


Рис. 190. Активный фильтр низких частот первого порядка

Пассивный аналог этого фильтра представлен на Рис. 191.

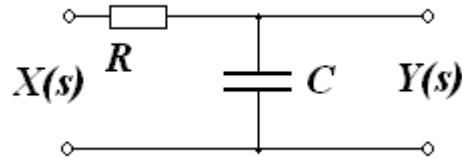


Рис. 191. Пассивный фильтр низких частот первого порядка

Коэффициент передачи фильтра равен

$$K(s) = -\frac{K_0}{1 + s\tau'_2}, \quad \tau'_2 = RC(K_0 + 1).$$

Для пассивного фильтра коэффициент передачи:

$$K(s) = -\frac{1}{1 + s\tau}, \quad \tau = RC$$

Частотные характеристики для пассивного фильтра приведены на Рис. 192 (для $\tau = 1$ с).

Сравнивая эти коэффициенты передачи, видим, что и в этом случае, как и для активных фильтров первого порядка высоких частот, при одинаковых постоянных времени τ'_2 и τ модуль коэффициента передачи активного фильтра первого порядка будет в K_0 раз больше, чем у пассивного.

Переходная характеристика фильтра Рис. 190 описывается выражением

$$h(t) = -\frac{K_0}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} \frac{1}{s(1 + s\tau'_2)} e^{st} ds = -K_0 \left(1 - e^{-\frac{t}{\tau'_2}} \right).$$

Это выражение также (с точностью до знака) совпадает с переходной характеристикой пассивного фильтра низких частот, которая изображена на Рис. 193.

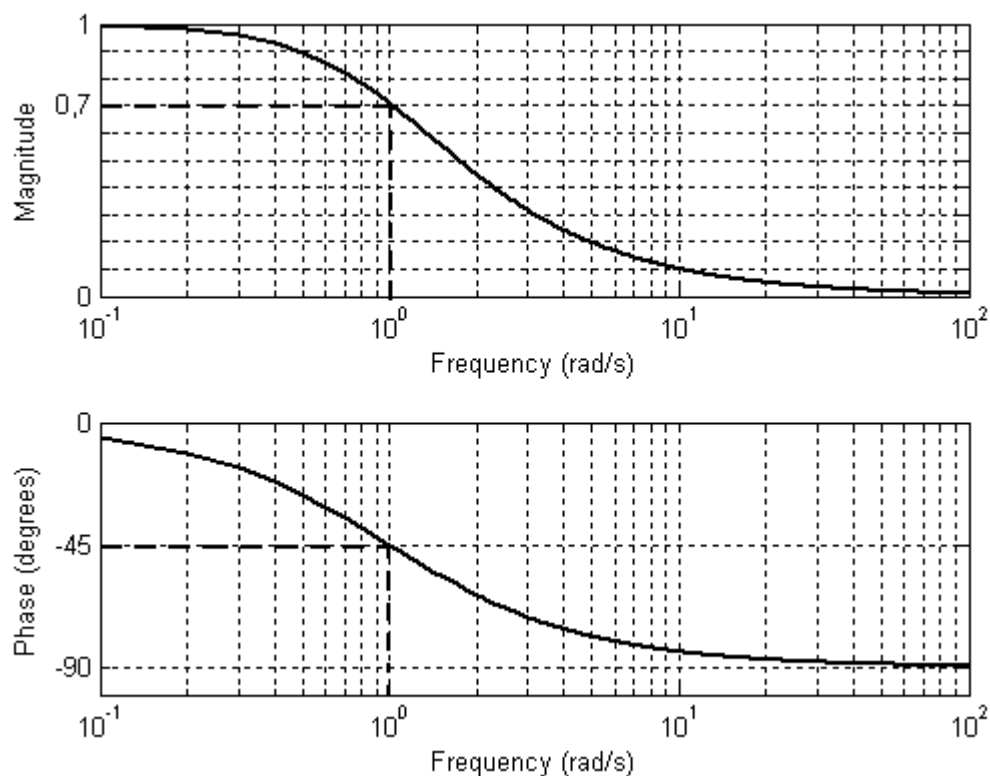


Рис. 192. АЧХ и ФЧХ пассивного фильтра низких частот (для $\tau = 1$ с)

Модель активного фильтра низких частот в SPS показана на Рис. 194. В приложении этой модели соответствует `filter2_low.mdl`.

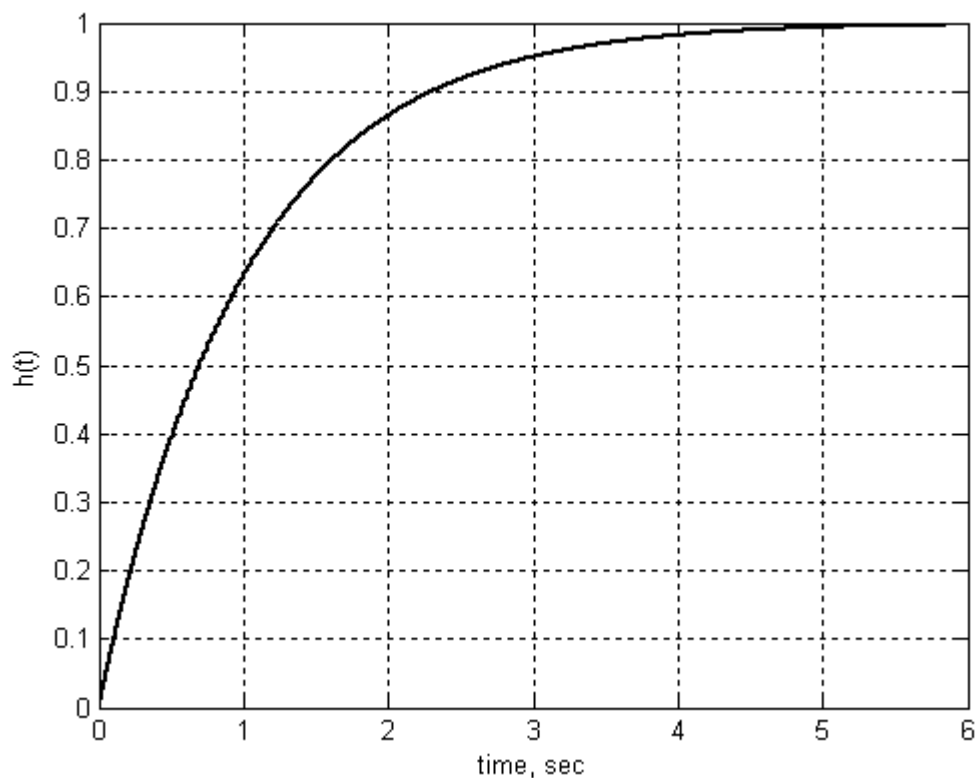


Рис. 193. Переходная характеристика пассивного фильтра низких частот (для $\tau = 1$ с)

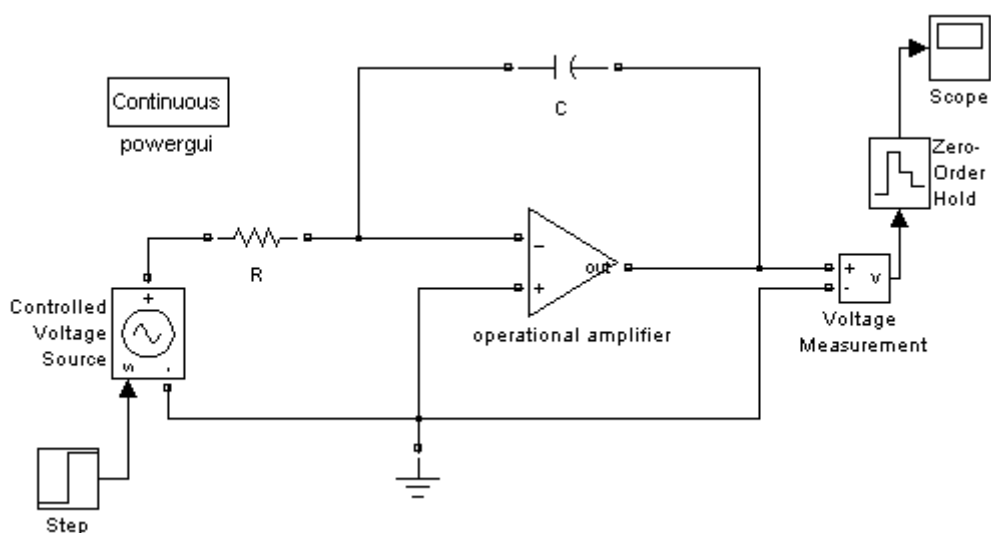


Рис. 194. Модель активного фильтра низких частот в SPS

Амплитудно-частотные характеристики активного фильтра низких частот (для $\tau = 1\text{с}$) показаны на Рис. 195.

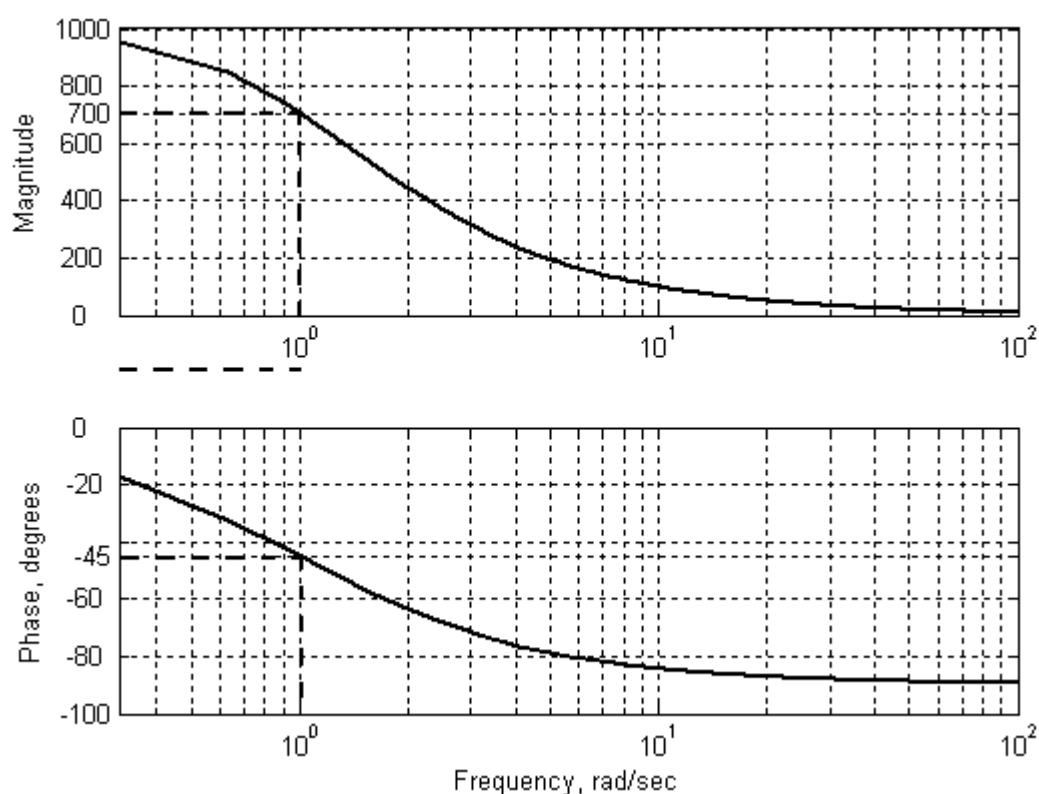


Рис. 195. АЧХ и ФЧХ активного фильтра низких частот (для $\tau = 1\text{с}$)

В заключение отметим, что учет конечного значения выходного сопротивления усилителя может в ряде случаев несколько изменить параметры активного фильтра, в частности его переходную характеристику. Покажем это на примере активного фильтра низких частот первого порядка. Схема, учи-

тая конечное значение выходного сопротивления $R_{\text{вых}}$, приведена на Рис. 196.

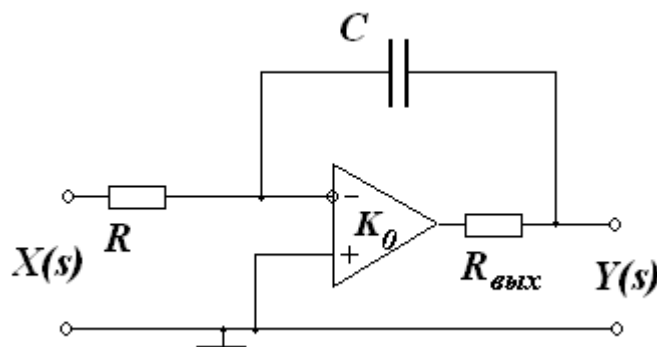


Рис. 196. Схема, учитывающая конечное значение выходного сопротивления $R_{\text{вых}}$

Можно показать, что коэффициент передачи в этом случае будет равен

$$K(s) = -\frac{K_0 - sR_{\text{вых}}C}{1 + sC[R(K_0 + 1) + R_{\text{вых}}]}$$

Эту функцию можно рассматривать как сумму коэффициентов передачи фильтров высокой и низкой частоты. Очевидно, что и переходная характеристика будет описываться соответствующей суммой. При моделировании, если необходимо, блок выходного сопротивления можно включить в подсистему операционного усилителя (Рис. 197).

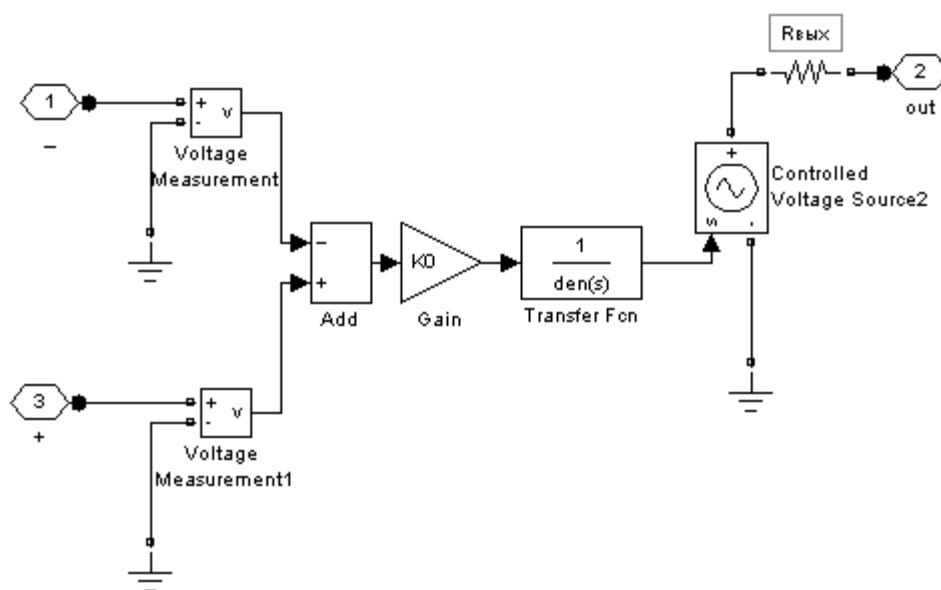


Рис. 197. Подсистема ОУ с выходным сопротивлением $R_{\text{вых}}$

Влияние конечного значения выходного сопротивления $R_{\text{вых}}$ на переходную характеристику фильтра иллюстрируется на Рис. 198. Эти графики

получены для тех же параметров активного фильтра низких частот первого порядка (для $\tau = 1$ с), но с разными выходными сопротивлениями ОУ. Из графика видно, что различие переходных характеристик пропадает уже через $t = 10^{-6}$ с.

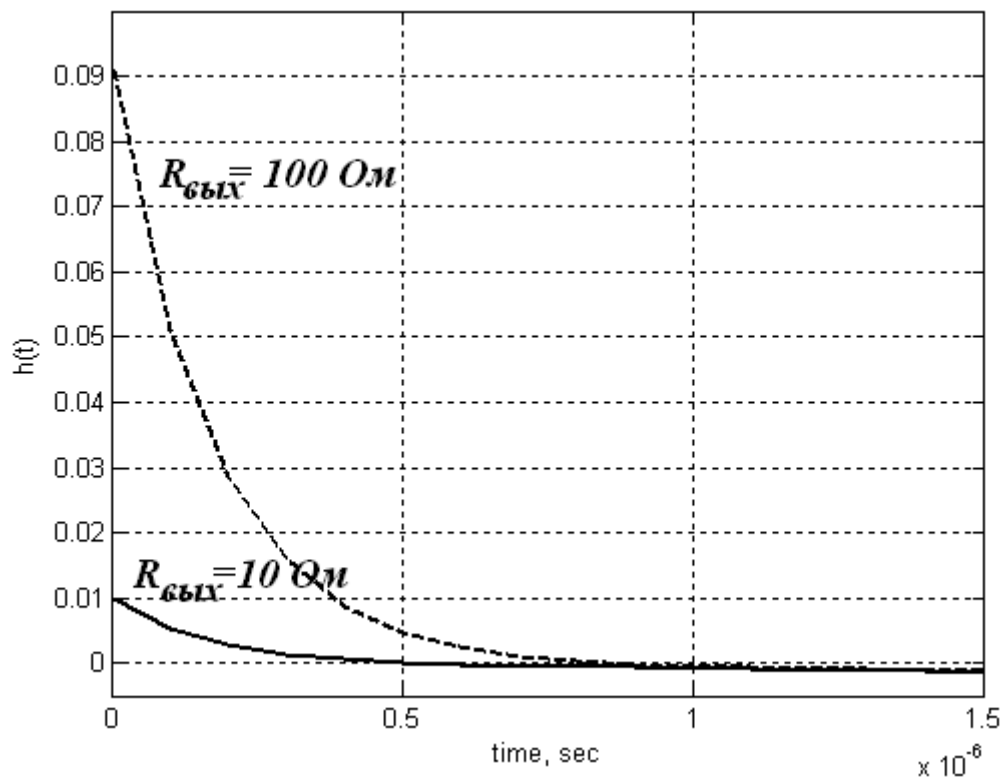


Рис. 198. Влияние конечного значения выходного сопротивления $R_{\text{вых}}$ на переходную характеристику фильтра

Отмеченные особенности активных фильтров выгодно отличают их от обычных дифференцирующих и интегрирующих RC-цепочек. Однако следует отметить ограничивающие обстоятельства при синтезе таких фильтров. При построении активных фильтров на высоких частотах становится заметной зависимость K_0 от частоты и принятая идеализация не выполняется. Поэтому приведенные выше характеристики на высоких частотах не реализуются. Другое ограничение проявляется при желании получить большие значения Q . Оказывается, что неизбежный разброс номиналов элементов и их нестабильность существенно сказываются на результирующей характеристике фильтра. Это весьма затрудняет настройку фильтра при $Q > 20$, и такие фильтры могут оказаться нестабильными в эксплуатации. Подобная чувствительность обусловлена тем, что для построения активного фильтра используется большее количество элементов. Так, для активного RC-фильтра второго порядка используется 5 элементов, тогда как RLC-фильтр содержит только 3 элемента.

Простые однокаскадные схемы практически используются только при $Q < 20$. Для построения активных фильтров большей добротности применя-

ются более сложные схемы. Их структура и методы синтеза рассмотрены в специальной литературе.

5.2. Усилители

Операционный усилитель на несколько порядков усиливает входной сигнал. С помощью цепи отрицательной обратной связи можно регулировать коэффициент усиления схемы. Для этого нужно включить некоторую пассивную электрическую цепь между выходом ОУ и его инвертирующим входом. Обратная связь также используется для обеспечения точности коэффициента усиления и временной стабильности.

5.2.1. Инвертирующий и неинвертирующий усилители

Если в схеме, изображенной на Рис. 133, заменить Z_1 и Z_2 резисторами R_1 и R_2 , то получится инвертирующий усилитель (входное напряжение подается на вход «-» ОУ), коэффициент усиления которого определяется по формуле из (23):

$$K = -\frac{Z_2}{Z_1} = -\frac{R_2}{R_1}$$

Входное сопротивление такого усилителя равно R_1 , т.к. именно на сопротивлении R_1 падает практически все напряжение сигнала $U_{вх}$.

Если входной сигнал подавать на вход «+» ОУ, то усилитель будет неинвертирующим (Рис. 199).

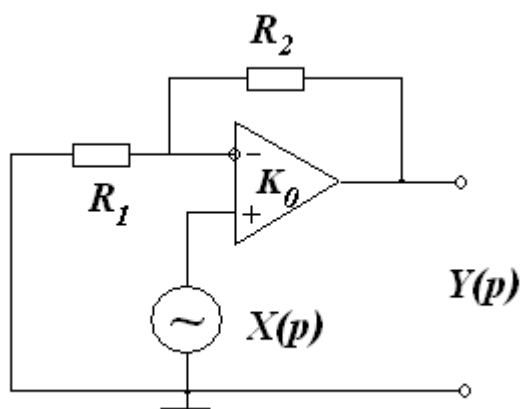


Рис. 199. Неинвертирующий усилитель

Неинвертирующий усилитель применяется в тех случаях, когда требуется усилить входной сигнал, сохранив его полярность. Рассмотрим Рис. 199. Усиливаемый сигнал подается на вход «+». Часть выходного сигнала через резистор обратной связи R_2 поступает на вход «-». Так как на вход «-» ОУ поступает сигнал той же полярности, что и на вход «+», разностный сигнал

оказывается меньше входного, что ведет к уменьшению усиления (см. также Рис. 132). Подробно принцип отрицательной обратной связи мы рассматривали в главе 5.1.2.

Из (20) выражение для коэффициента передачи усилителя с отрицательной обратной связью будет

$$\dot{K} = \frac{\dot{K}_0}{1 + \beta \dot{K}_0}. \quad (59)$$

Из этого выражения следует, что от свойств цепи обратной связи существенно зависит величина коэффициента передачи усилителя. Коэффициент передачи цепи обратной связи вычисляется по формуле

$$\beta = \frac{R_1}{R_1 + R_2}$$

и, следовательно, учитывая в (59), что коэффициент усиления ОУ K_0 очень большой, можно получить выражение для коэффициента передачи усилителя

$$\dot{K} = \frac{\dot{K}_0}{1 + \beta \dot{K}_0} \approx \frac{1}{\beta} = \frac{R_2}{R_1} + 1. \quad (60)$$

Заметим, что коэффициент передачи в данном случае не зависит от частоты.

Рассмотрим модель неинвертирующего усилителя в **SPS**, изображенную на Рис. 200 (в приложении модель **amplifier.mdl**). Левая часть рисунка представляет электрическую схему, а правая часть – обработку полученного сигнала. В электрической схеме на «+» вход ОУ (**operational amplifier**) подается синусоидальное напряжение амплитудой 0,1 В и частотой 10 Гц (блок **AC Voltage Source**). Цепь обратной связи состоит из реостатов $R_1 = 10^4$ Ом и $R_2 = 50 \cdot 10^4$ Ом. Из уравнения (60) можно рассчитать, что коэффициент усиления такого усилителя будет равен 51. Для того чтобы получить точное значение амплитуды сигнала на выходе использовались четыре блока, изображенных справа на Рис. 200. С помощью блока **Zero-Order Hold** сигнал оцифровывался с частотой квантования $f_s = 10^{-3}$.

С помощью блока **Buffer** производится «накопление» сигнала. Этот блок перераспределяет поступающую на него последовательность чисел в вектор или фрейм (**frame**), размер которого задается в параметрах блока. Сигнал на выходе блока «буфер» представляет собой последовательность фреймов, промежуток времени между которыми (период квантования) увеличивается пропорционально увеличению размеров фрейма (см. Рис. 201). С помощью блока «буфер» можно выполнять и обратную операцию: понижать размерность сигнала и, соответственно, уменьшать период квантования (для подробностей и других возможностей блока см. справку **MATLAB**).

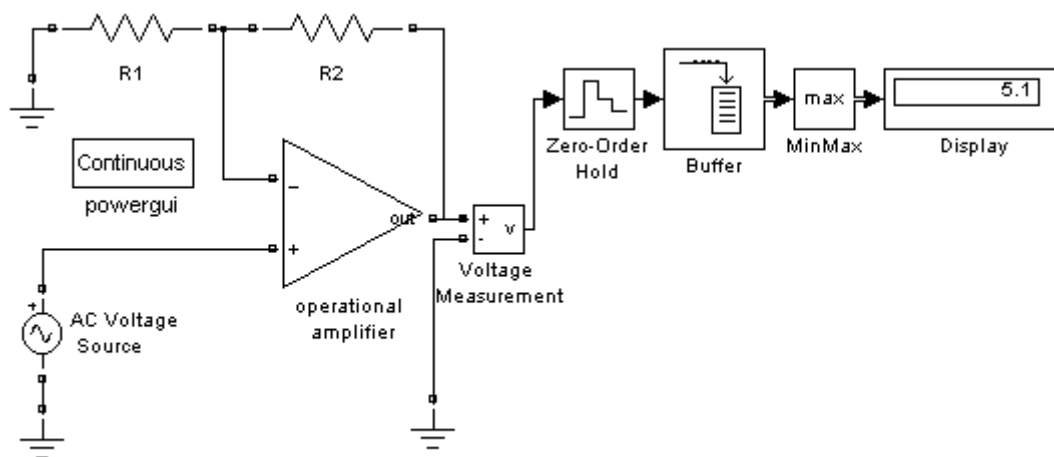


Рис. 200. Модель неинвертирующего усилителя в SPS

Мы используем данный блок для того, чтобы получить один фрейм из всего сигнала длительностью $time$ (время моделирования). Поэтому в параметрах блока «буфер» в окне **Output buffer size** (размер фрейма) установим значение, соответствующее $time/f_s$, в окнах **Buffer overlap** (перекрывание фреймов) и **Initial conditions** (начальные условия) поставим нули. Заметим, что соединяющая блоки линия после блока «буфер» стала двойной, напоминая, что размерность сигнала изменилась (Рис. 200).

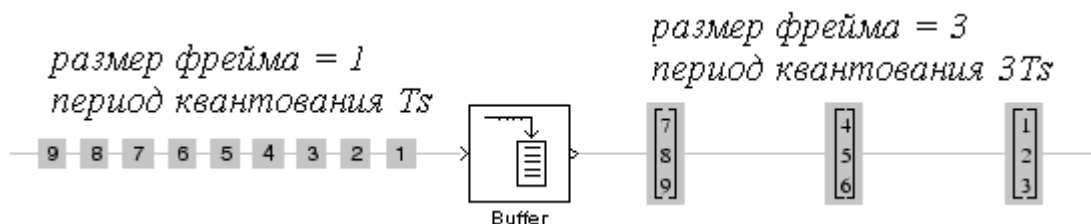


Рис. 201. Пояснение к работе блока «буфер»

Далее идет блок **MinMax**, выбирающий из фрейма максимальное значение, и потом это число отображается с помощью блока **Display**. Для идеального ОУ (при выбранных нами значениях R_1 и R_2 и амплитуде входного сигнала) амплитуда выходного сигнала должна быть равна 5,1 В. Однако это значение заметно изменяется, если вводить в модель «нелинейности по амплитуде и частоте», выходные и входные сопротивления или уменьшать коэффициент усиления ОУ. На Рис. 202 изображена упрощенная модель ОУ (подсистема), которая использовалась при расчетах.

Для достижения «идеальности» входные сопротивления брались достаточно большими 10^8 Ом. Так как на «+» вход ОУ сигнал подавался непосредственно с источника напряжения, то убрать из модели ОУ входное сопротивление по неинвертирующему входу было нельзя (источник напряжения замыкался на идеальный вольтметр).

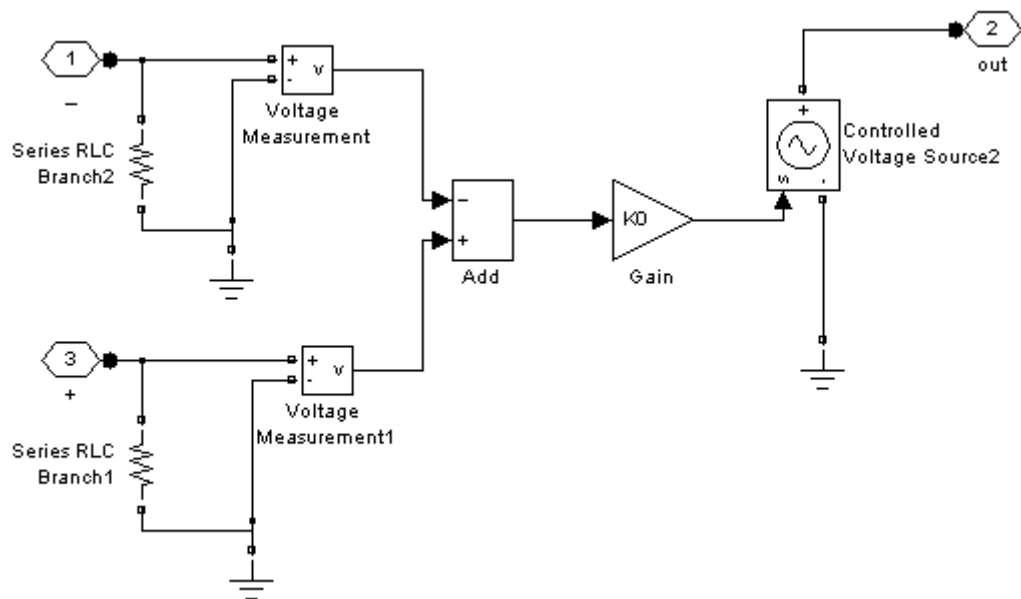


Рис. 202. Модель ОУ

Рассмотрим схему, в которой это можно избежать. На Рис. 203 изображен неинвертирующий усилитель, на «+» вход которого подается сигнал, прошедший через фильтр высоких частот (чтобы не пропустить постоянное напряжение).

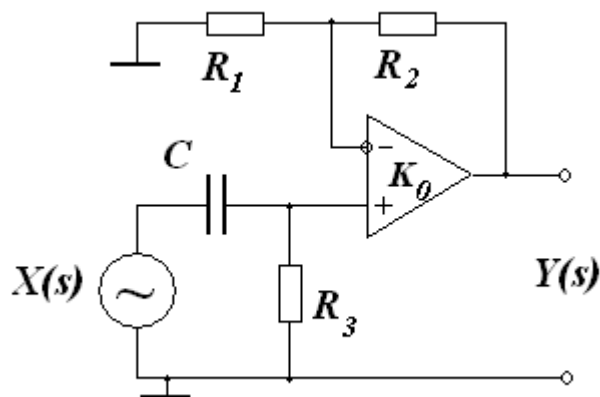


Рис. 203. Неинвертирующий усилитель переменного напряжения

Эта схема является по существу неинвертирующим усилителем, ее коэффициент усиления K на средних и высоких частотах (там где коэффициент передачи фильтра высоких частот равен 1) рассчитывается по формуле (60). Нижний предел полосы пропускания определяется величиной емкости C и сопротивлением R_3 , т.е. граничной частотой фильтра высоких частот на входе ОУ:

$$f_H = \frac{\omega}{2\pi} = \frac{1}{2\pi\tau} = \frac{1}{2\pi CR_3}.$$

Верхний предел полосы пропускания определяется свойствами самого ОУ, емкостью и сопротивлением нагрузки и выходным сопротивлением ОУ.

На Рис. 204 изображена модель этого усилителя. В приложении этой схеме соответствует модель **amplifier_2.mdl**. Сопротивления R_1 и R_2 остались прежними 10^4 Ом и $50 \cdot 10^4$ Ом соответственно. C и R_3 выберем так, чтобы $f_n = 100$ Ом ($C = 100$ мкФ, $R_3 = 10^3/2\pi$). На вход от источника **Chirp** подается синусоидальный сигнал с линейно изменяющейся частотой и амплитудой 0,1 В. Частота меняется от 0 до 8 кГц за промежуток времени 20 мс (это совпадает с временем моделирования).

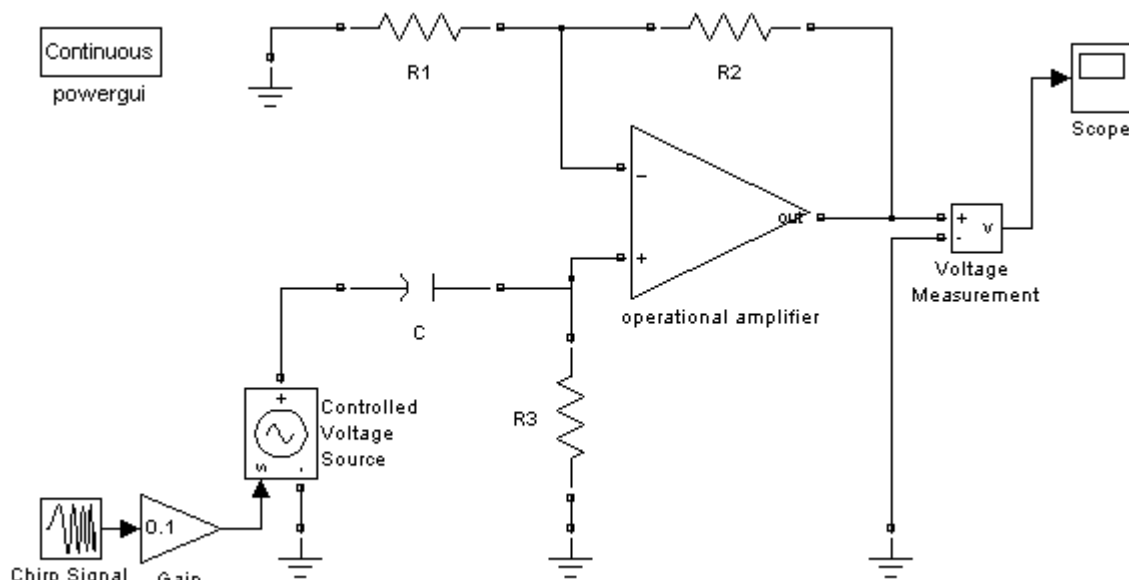


Рис. 204. Модель неинвертирующего усилителя переменного напряжения

Напряжение на выходе этой схемы показано на Рис. 205. К моменту времени $t = 10$ мс частота сигнала становится равной 4 кГц, граничной частоты сигнал достигает при $t = 100 \text{ Гц} \cdot 20 \text{ мс} / 8 \text{ кГц} = 0,25 \text{ мс}$.

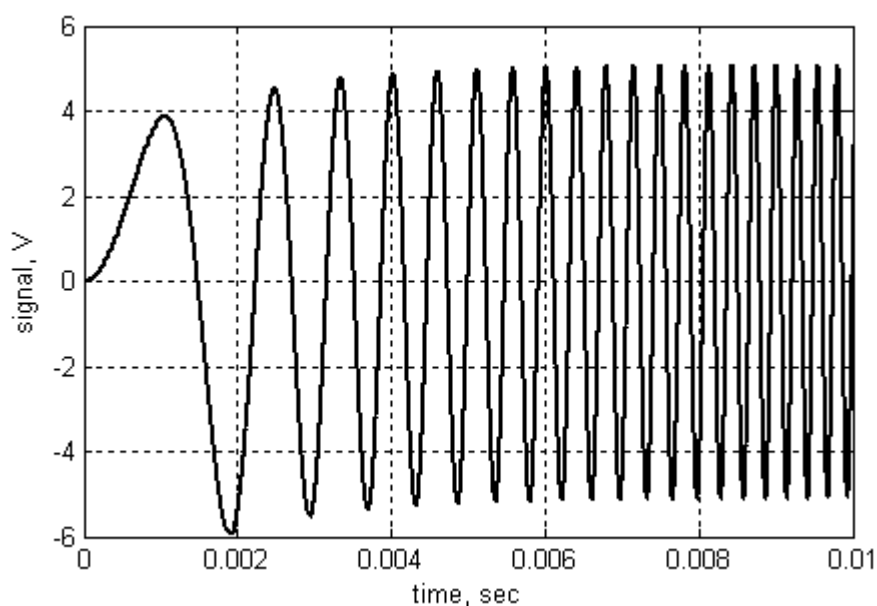


Рис. 205. Сигнал на выходе неинвертирующего усилителя

В некоторых случаях не столь важным является усиление по напряжению, как способность усилителя согласовывать высокое внутреннее сопротивление источника сигналов с низким и, возможно, изменяющимся сопротивлением нагрузки. Для этих целей используется повторитель напряжения (Рис. 206).

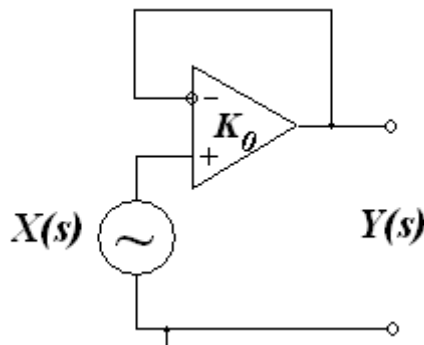


Рис. 206. Повторитель напряжения

Это по существу неинвертирующий усилитель, в котором R_1 заменено бесконечным сопротивлением, R_2 – нулевым сопротивлением. Очевидно, что коэффициент передачи такой схемы равен 1, входное сопротивление велико, а выходное мало.

5.3. Генераторы сигналов на ОУ

Генераторами называются электронные схемы, формирующие переменное напряжение требуемой формы. ОУ можно включить таким образом, что они будут работать как генераторы сигналов. Можно получить, например, сигналы прямоугольной, треугольной, пилообразной и синусоидальной формы. Рассмотрим сначала генераторы синусоидальной формы, а затем генераторы сигналов специальной формы, в частности, генераторы прямоугольного и треугольного напряжения.

5.3.1. Условия генерации

Рассмотрим условия, при которых будет происходить генерация на какой-либо определенной частоте – условия возбуждения. На Рис. 207 изображена блок-схема генератора.

ОУ усиливает входной сигнал в K_0 раз. При этом между входным $U_{\text{вх}}$ и выходным U_1 напряжениями усилителя возникает паразитный фазовый сдвиг φ_1 . К выходу усилителя подключена схема частотно-зависимой обратной связи, которая может представлять собой, например, колебательный контур.

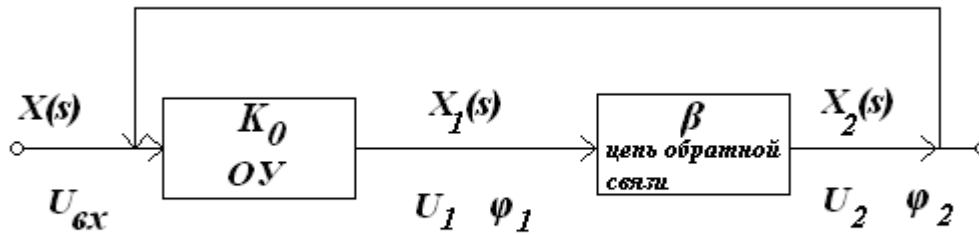


Рис. 207. Блок-схема генератора

По определению, коэффициент передачи обратной связи равен $\beta(s) = \frac{X_2(s)}{X_1(s)}$. Обозначим фазовый сдвиг между напряжениями U_1 и U_2 как φ_2 .

Условием генерации замкнутой системы является равенство выходного напряжения схемы обратной связи и входного напряжения усилителя. Это условие можно записать следующим образом:

$$X(s) = X_2(s) = \beta(s)X_1(s) = \beta(s)K_0X(s).$$

Отсюда

$$\beta(s)K_0 = 1.$$

Из этого соотношения следуют два условия для амплитуд и фаз комплексных величин:

$$|\beta||K_0| = 1 \text{ — условие баланса амплитуд и}$$

$$\varphi_1 + \varphi_2 = 2\pi k \text{ (где } k \text{ — целое число) — условие баланса фаз.}$$

Условие баланса амплитуд заключается в том, что схема генератора может возбуждаться только тогда, когда усилитель компенсирует потери в схеме обратной связи. Условие баланса фаз означает, что колебания в замкнутой системе могут возбуждаться только тогда, когда фаза выходного напряжения схемы обратной связи и фаза входного напряжения усилителя совпадают.

5.3.2. LC-генератор

В качестве примера рассмотрим LC-генератор (Рис. 208).

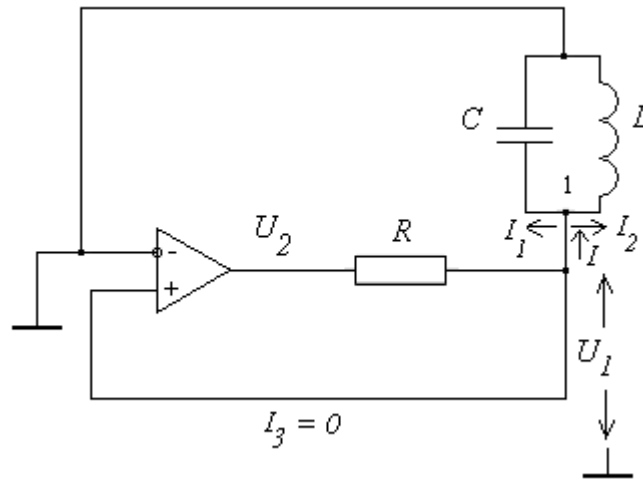


Рис. 208. LC-генератор

ОУ, включенный по неинвертирующей схеме, усиливает входное напряжение U_1 в K_0 раз. Так как ОУ имеет небольшое выходное сопротивление, то параллельный колебательный контур схемы обратной связи подключается к нему через резистор R . Напряжение U_2 на выходе ОУ, как обычно, рассматривается относительно «земли». Учитывая большое входное сопротивление ОУ, примем I_3 ток положительной обратной связи равным нулю. Для определения влияния параметров цепи обратной связи на входное напряжение применим для точки 1 схемы первый закон Кирхгофа:

$$I - I_1 - I_2 = 0$$

$$\frac{U_2 - U_1}{R} - C \frac{dU_1}{dt} - \frac{1}{L} \int U_1 dt = 0$$

так как $U_2 = K_0 U_1$, уравнение примет вид

$$\frac{d^2 U_1}{dt^2} + \frac{(1 - K_0)}{RC} \frac{dU_1}{dt} + \frac{1}{LC} U_1 = 0.$$

Это дифференциальное уравнение квазигармонических колебаний. Введем обозначения

$$\gamma = \frac{(1 - K_0)}{2RC}$$

$$\omega_0^2 = \frac{1}{LC}$$

Тогда дифференциальное уравнение примет вид:

$$\frac{d^2 U_1}{dt^2} + 2\gamma \frac{dU_1}{dt} + \omega_0^2 U_1 = 0.$$

Это уравнение имеет следующее решение:

$$U_1(t) = U_0 e^{-\gamma t} \sin\left(\sqrt{\omega_0^2 - \gamma^2} t + \varphi_0\right).$$

Константы U_0 и φ_0 находятся из начальных условий.

Различают три характерных случая:

1. $\gamma > 0$, когда $K_0 < 1$ (не выполняется для ОУ)

Амплитуда выходного сигнала падает по экспоненциальному закону, колебания затухающие (Рис. 209). Жирной линией на рисунке показана огибающая сигнала.

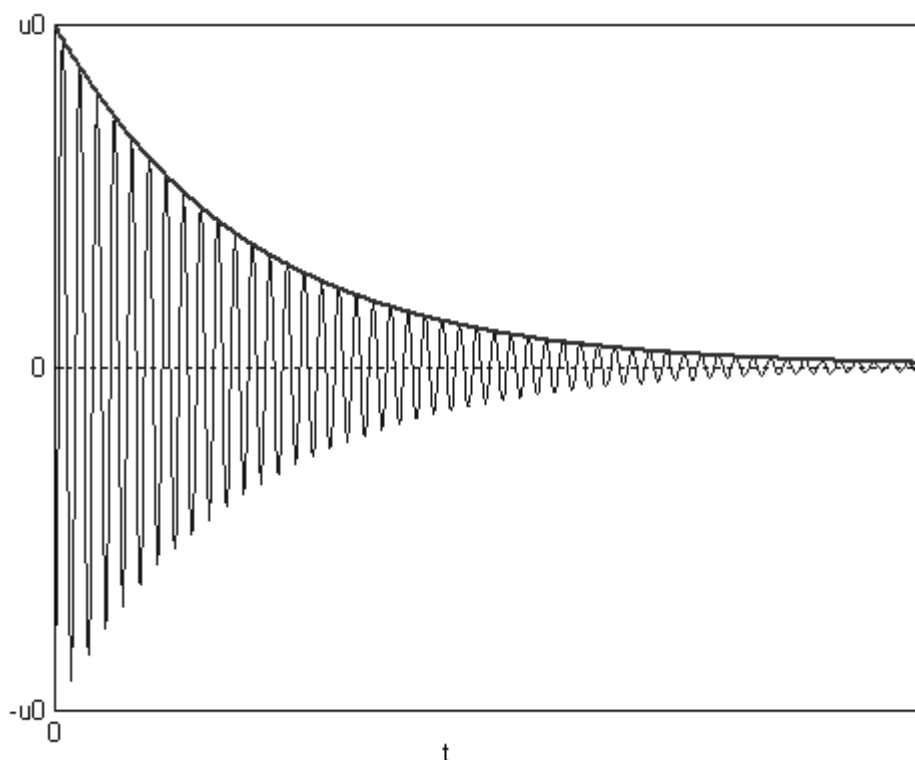


Рис. 209. Затухающие колебания

2. $\gamma = 0$, когда $K_0 = 1$ (также не выполняется для ОУ)

Возникают синусоидальные колебания с частотой $\omega_0 = \frac{1}{\sqrt{LC}}$ и постоянной амплитудой.

3. $\gamma < 0$, когда $K_0 > 1$

Амплитуда выходного напряжения возрастает по экспоненциальному закону (Рис. 210).

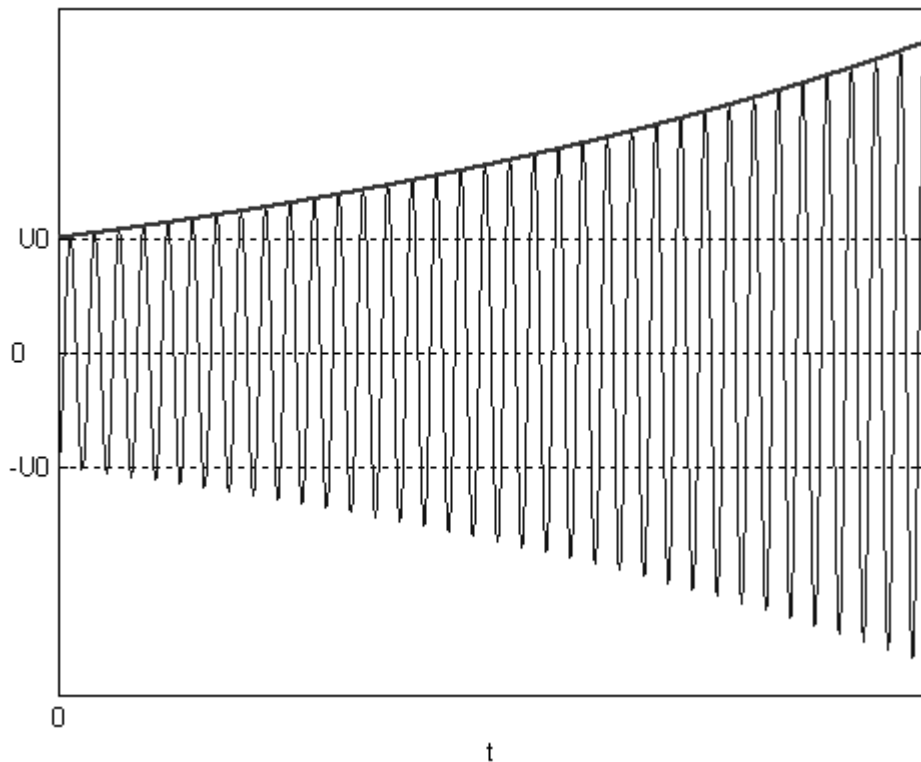


Рис. 210. Колебания с увеличивающейся амплитудой

Так как K_0 ОУ намного больше единицы, то при включении питания будет происходить самовозбуждение, амплитуда выходного напряжения будет возрастать, пока усилитель не достигнет режима насыщения (U_{max}). После этого будут происходить колебания с постоянной амплитудой. Рассмотрим схему в SPS (Рис. 211). Название модели в приложении **LC_generator.mdl**.

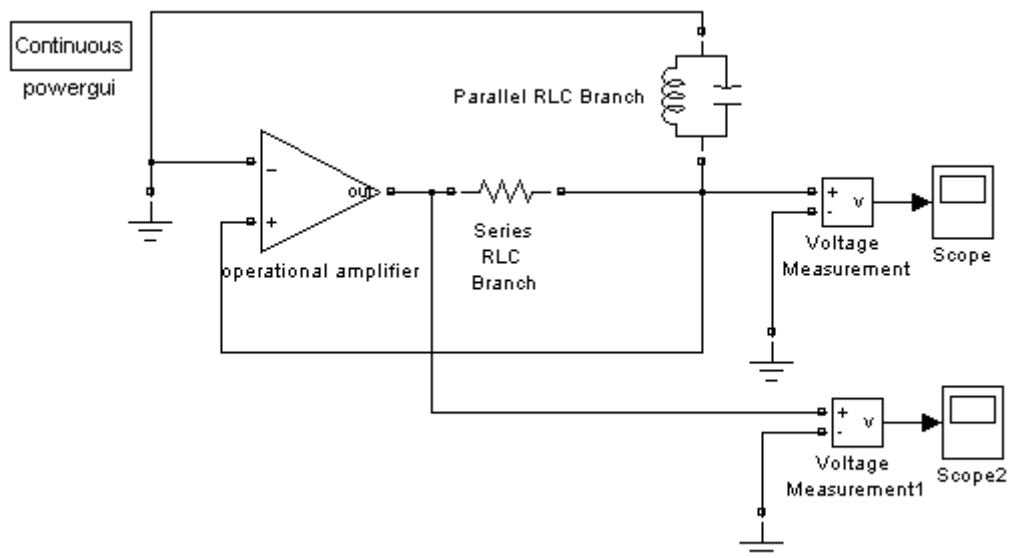


Рис. 211. Модель LC-генератора в SPS

Для начала генерации необходимо подать напряжение на вход ОУ. В реальной схеме эту роль может выполнять тепловой шум на входе ОУ, уси-

ливаемый в десятки тысяч раз. Для моделирования мы выберем начальное напряжение на конденсаторе в колебательном контуре. Пусть это значение будет очень мало. На Рис. 212 изображено окно параметров блока колебательного контура. В нижней части окна устанавливается начальное значение напряжения на конденсаторе. Нужно установить флажок **Set the initial capacitor voltage** и ввести значение в окно **Capacitor initial voltage** 10^{-5} В

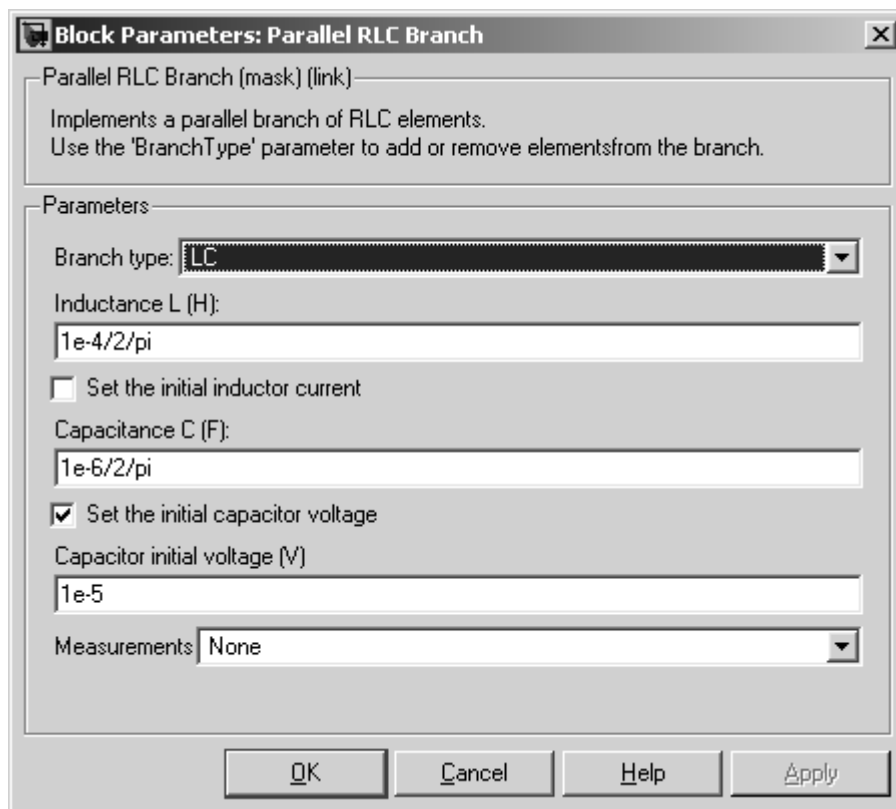


Рис. 212. Окно параметров блока колебательного контура

Из выбранных значений индуктивности (окно **Inductance** на Рис. 212) и емкости (окно **Capacitance** на этом рис.) катушки и конденсатора, составляющих колебательный контур, можно определить частоту колебаний.

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi\sqrt{LC}} = \frac{1}{2\pi\sqrt{\frac{10^{-4}}{2\pi} \frac{10^{-6}}{2\pi}}} = 10^5 \text{ Гц}$$

Сопротивление реостата (**Series RLC Branch** на Рис. 211) выберем $R = 10^4$ Ом, а параметры ОУ – коэффициент усиления 1000 и пределы ± 15 В.

На Рис. 213 изображено напряжение на выходе ОУ, регистрируемое вторым осциллографом на Рис. 211. Видно, что после достижения режима насыщения, операционный усилитель выступает в роли генератора прямоугольных импульсов с частотой равной собственной частоте колебательного контура f_0 .

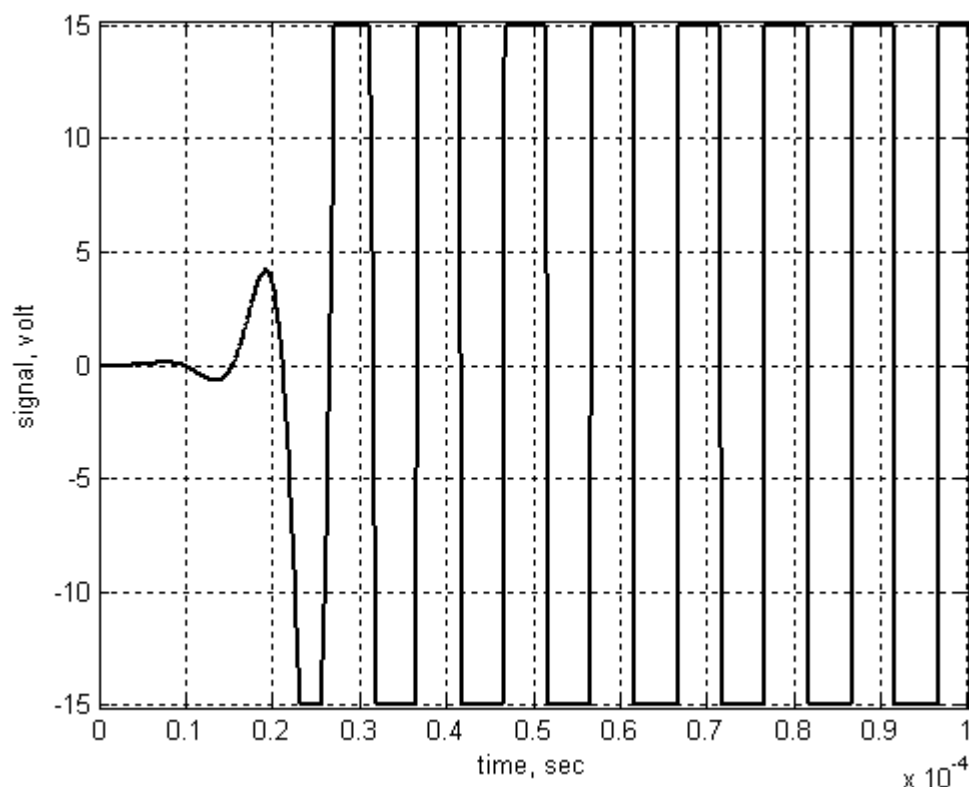


Рис. 213. Сигнал на выходе ОУ

Поэтому, схему Рис. 211 можно заменить эквивалентной с генератором прямоугольных импульсов той же частоты и амплитуды (в приложении модель **LC_generator_equivalent.mdl**).

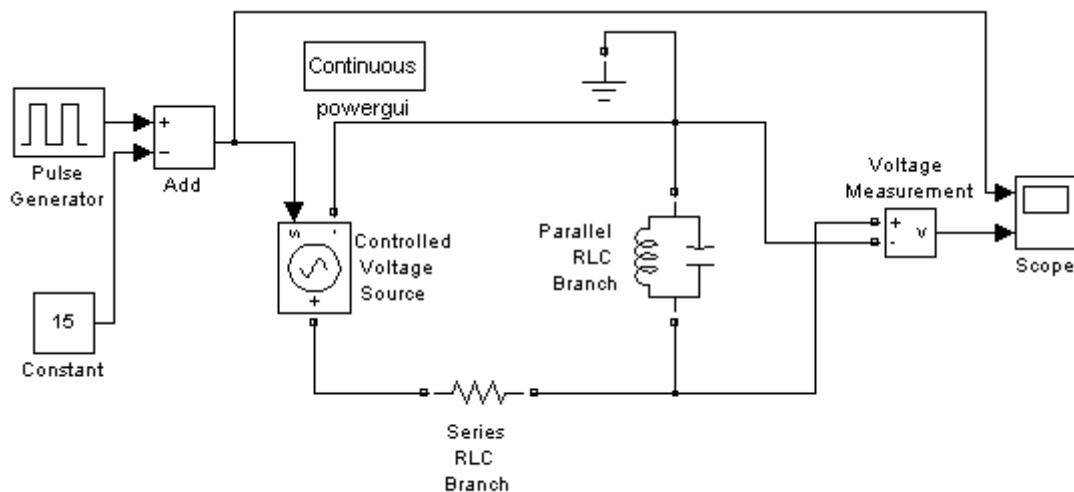


Рис. 214. Эквивалентная схема

Амплитуда и форма сигнала на выходе может быть разной и будет зависеть от параметров колебательного контура и величины сопротивления. На Рис. 215 изображен внешний вид сигнала на выходе генератора (для того, чтобы на графике различить колебания, частота сигнала была уменьшена в 30 раз).

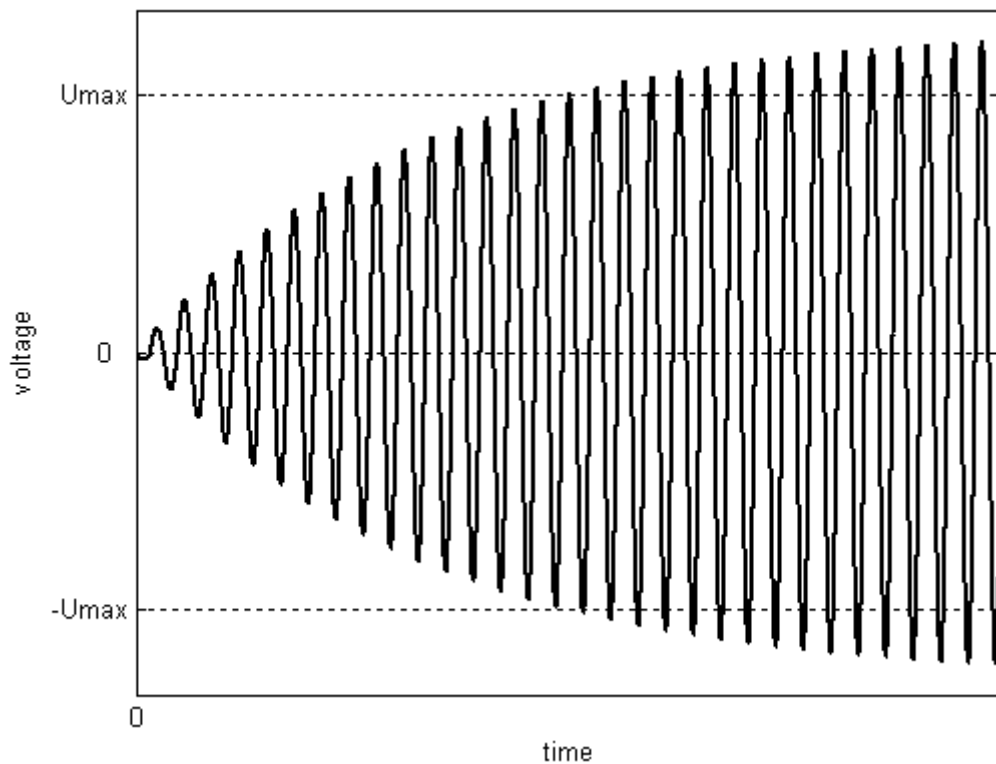
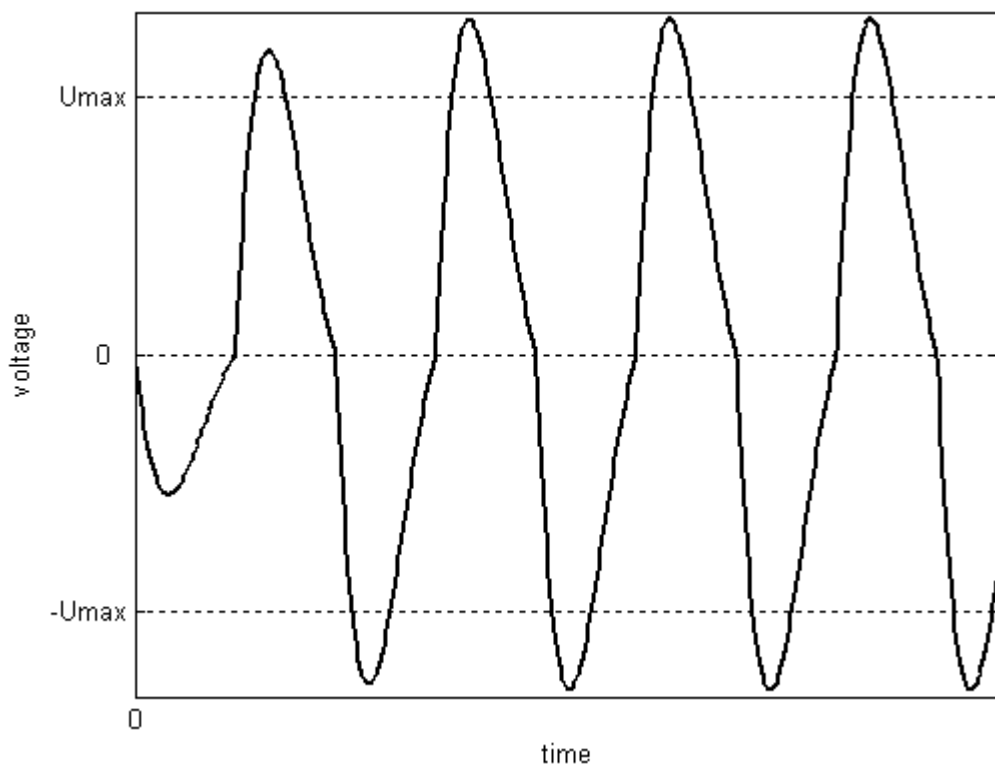


Рис. 215. Сигнал на выходе генератора

При этом форма выходного сигнала будет отличаться от синусоиды. Это особенно становится заметно, если, например, изменять сопротивление R , не изменяя собственную частоту колебательного контура. На Рис. 216 показана различная форма сигнала ($R = 10$ Ом (сверху) и $R = 1$ Ом (снизу)).



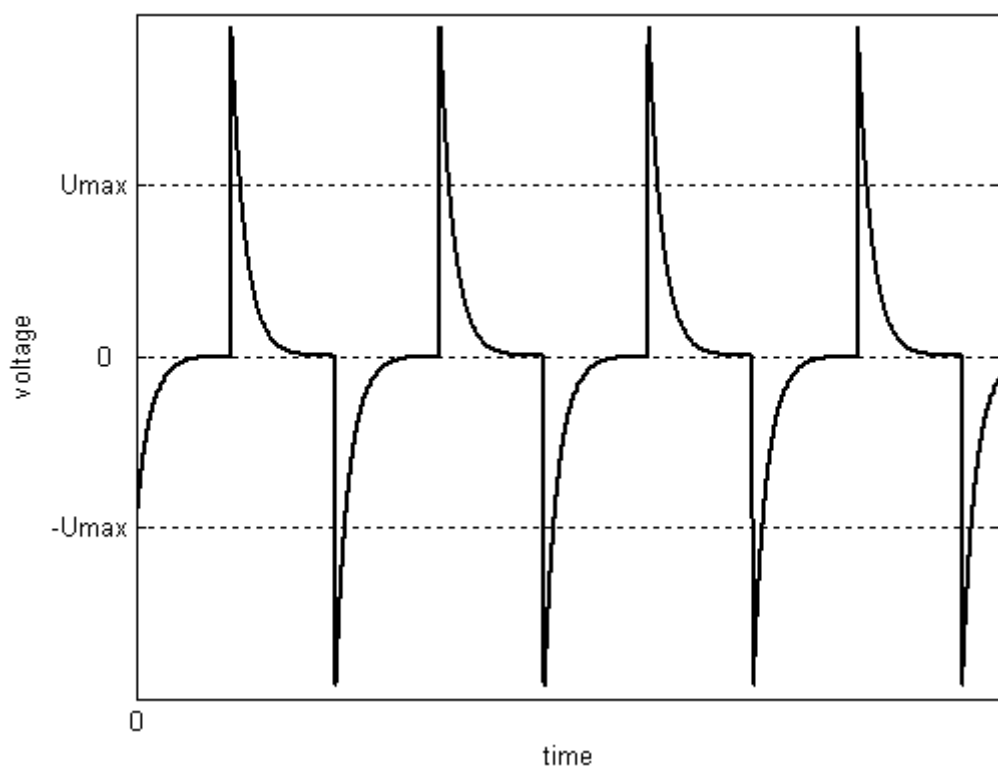


Рис. 216. Варианты сигналов на выходе генератора

Если требуется синусоидальное выходное напряжение, необходимо осуществить регулировку коэффициента усиления таким образом, чтобы он стал равным единице (для обеспечения отсутствия насыщения усилителя). Для этого в цепь отрицательной обратной связи нужно включить делитель напряжений. На Рис. 217 изображена схема Рис. 208, дополненная делителем напряжений.

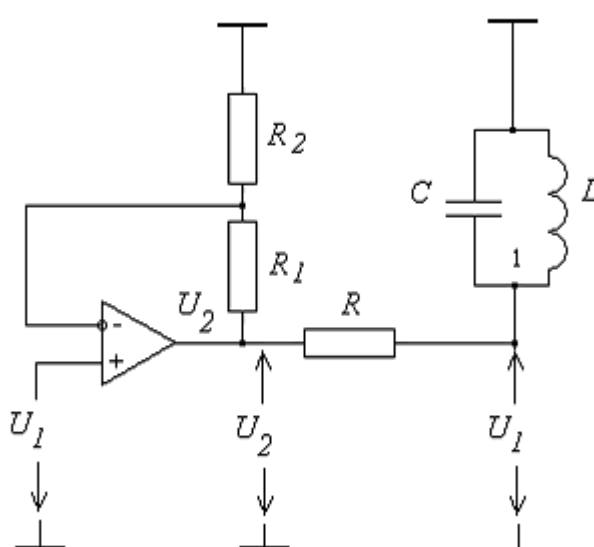


Рис. 217. Схема LC -генератора с делителем напряжений

Из условий $U_2=U_1$ (коэффициент усиления равен единице), $U_2=K_0(U_1-U_{R1})$ – дифференциальный усилитель и $U_2=U_{R1}+U_{R2}$ можно получить отношение сопротивлений: $\frac{U_{R1}}{U_{R2}} = \frac{R_1}{R_2} = (K_0 - 1)$.

На высоких частотах довольно легко можно реализовать колебательный контур с высокой добротностью. При этом напряжение на колебательном контуре даже при глубоком насыщении усилителя остается практически синусоидальным. Поэтому в схемах высокочастотных генераторов обычно не применяют специальных методов регулирования амплитуды выходного сигнала усилителя, а выходное напряжение снимают непосредственно с колебательного контура.

5.3.3. Генератор с мостом Вина

LC -генераторы почти не используются для получения низкочастотных колебаний, так как при этом требуются большие величины индуктивности и емкости. В этом частотном диапазоне преимущественно используются генераторы, частота генерации которых определяется параметрами RC -фильтров. Например, в цепь положительной обратной связи можно включить цепочку Вина (Рис. 218).

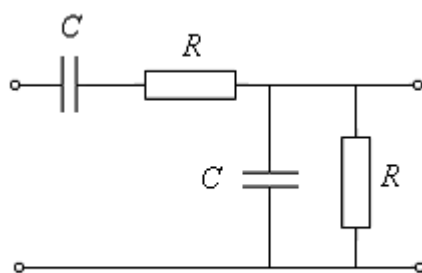


Рис. 218. Цепочка Вина

Это избирательный фильтр второго порядка. В параграфе 2.11.3 было выведено выражение для коэффициента передачи этой цепочки и рассмотрены ее частотные характеристики. Было получено, что максимальный коэффициент передачи цепочки Вина равен $1/3$ на частоте $f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi RC}$, а сдвиг фаз на этой частоте равен нулю.

Для устойчивой генерации монохроматического колебания необходимо выполнить условия баланса амплитуд и фаз. Так как коэффициент передачи цепочки Вина на частоте резонанса составляет $1/3$ при нулевом сдвиге фаз, то коэффициент усиления схемы включения ОУ должен быть равен 3. На Рис. 219 показано включение входов ОУ в диагональ моста Вина. Делитель напряжения, включенный в инвертирующую обратную цепь, обеспечивает коэффициент усиления схемы включения ОУ равный 3. Переменный резистор

R_7 включен для регулировки, т.к. для устойчивой генерации необходимо точно выполнить приведенные выше условия, на практике же сопротивления и емкости имеют некоторое отклонение от номиналов.

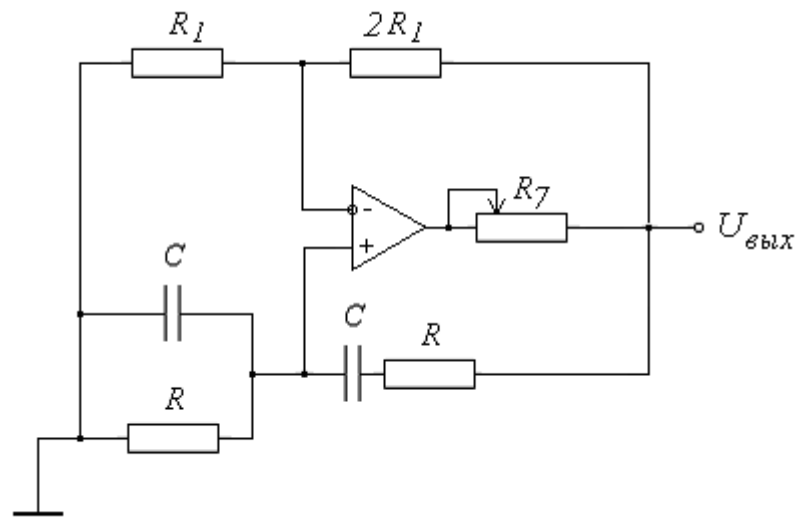


Рис. 219. Схема генератора с мостом Вина

Если коэффициент усиления схемы включения ОУ будет больше трех, то сигнал будет ограничиваться и иметь трапецевидную форму (Рис. 220).

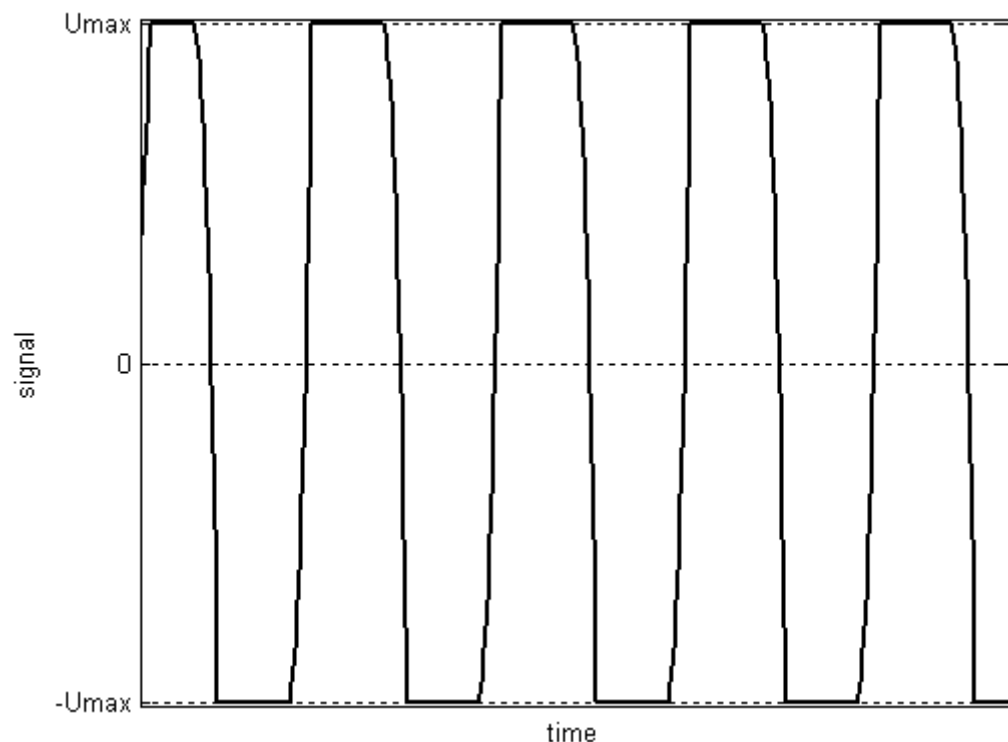


Рис. 220. Искаженный генерируемый сигнал при $K_0 > 3$

Если же коэффициент усиления будет меньше трех, то колебания постепенно погаснут. Рассмотрим результаты работы модели этого генератора

(Рис. 221). Схема модели отличается от схемы, изображенной на Рис. 219 только отсутствием резистора R_7 , который не нужен при моделировании.

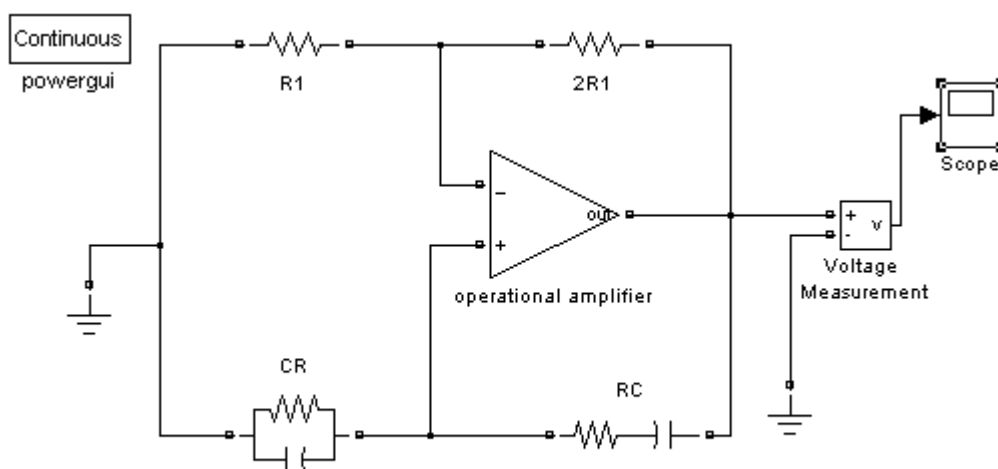


Рис. 221. Модель генератора с мостом Вина в SPS

Для возникновения генерации, в качестве начальных условий выбрано начальное напряжение на конденсаторе параллельной CR цепочки, равное 1 В. Значения параметров выбраны так, чтобы частота генерации была 1 кГц ($R=10^3/2\pi$ Ом и $C=10^{-6}$ Ф). Полученный сигнал приведен на Рис. 222.

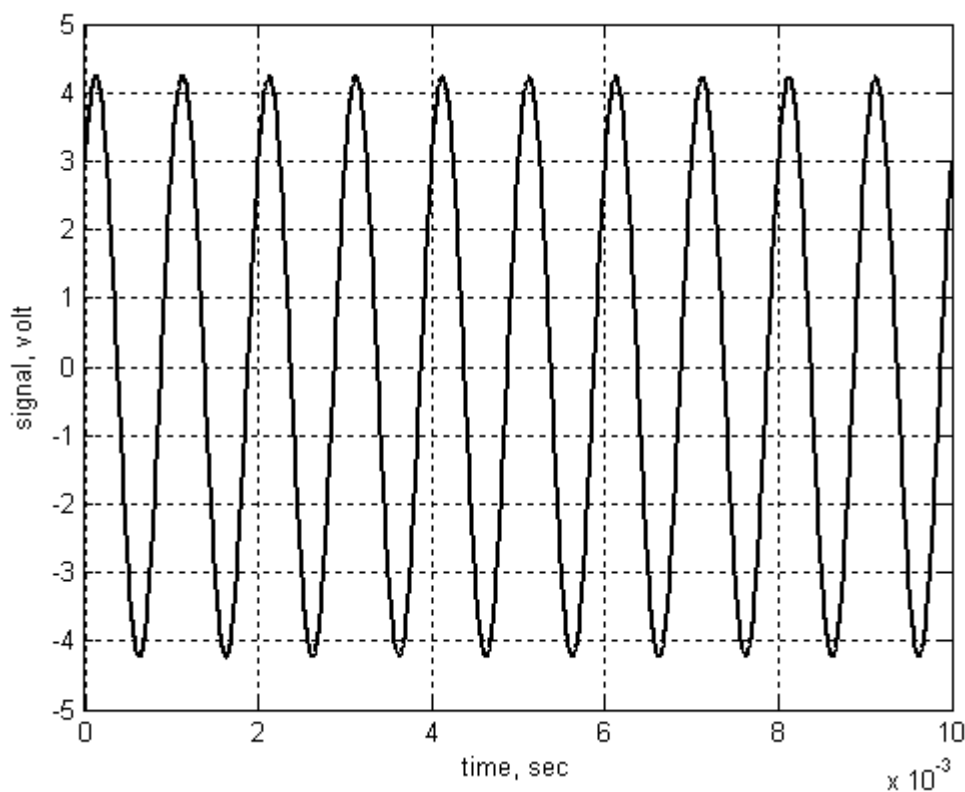


Рис. 222. Генерируемый сигнал при $K_0 = 3$

Амплитуда сигнала определяется начальным значением на конденсаторе, но не выходит за пределы напряжения насыщения ОУ. На Рис. 223 представлен случай, когда начальное значение напряжения на конденсаторе равно 20В.

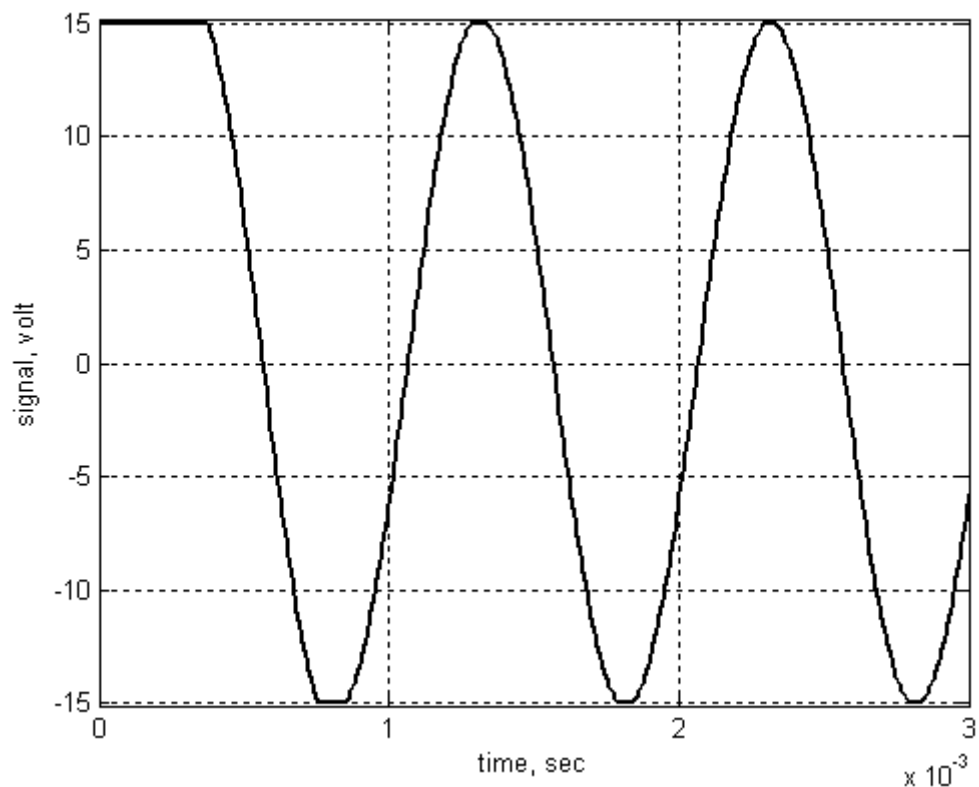


Рис. 223. Генерируемый сигнал при большом начальном значении

5.3.4. Генератор импульсных сигналов

Рассмотрим принцип работы генератора колебаний прямоугольной формы (Рис. 224). Такую схему обычно называют мультивибратором.

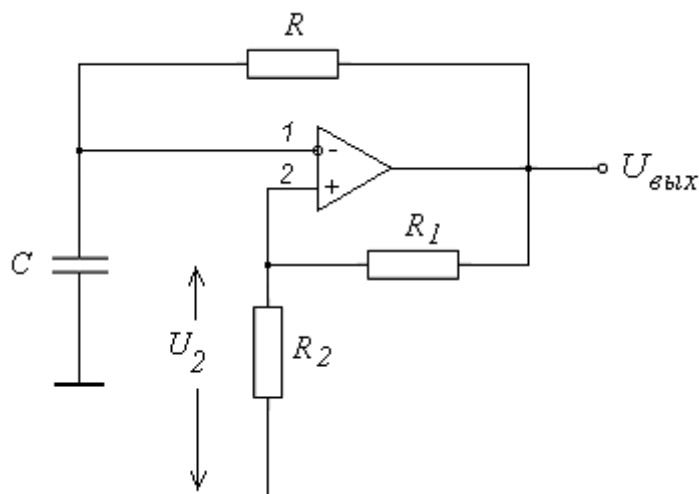


Рис. 224. Мультивибратор

В момент подачи напряжений питания $+U$ и $-U$ на инвертирующий вход 1 напряжение с первоначально разряженного конденсатора C не поступает, т.е. вход 1 в начальный момент времени заземлен. Однако в тот же момент на сопротивлении R_2 появляется небольшое положительное или отрицательное напряжение U_2 , и это напряжение прикладывается к неинвертирующему входу 2. Напряжение U_2 появляется в начальный момент, поскольку на выходе имеется положительное или отрицательное напряжение сдвига $U_{\text{вых. сдвига}}$ даже в том случае, когда ко входам 1 и 2 не приложено дифференциального входного напряжения.

Резисторы R_1 и R_2 образуют делитель напряжения, и часть выходного напряжения падает на R_2 . Так как инвертирующий вход 1 первоначально заземлен через разряженный конденсатор C , то в начальный момент времени все напряжение U_2 выделяется между входами 1 и 2. Даже если U_2 мало, оно начнет переводить ОУ в режим насыщения, т.к. если напряжение выходного сдвига $U_{\text{вых. сдвига}}$ положительно, то напряжение U_2 на неинвертирующем входе 2 также положительно. Это напряжение U_2 сразу усиливается в число раз, равное коэффициенту усиления ОУ без обратной связи K_0 , и приводит выходное напряжение к его предельному значению $U_{\text{вых. макс.}}$, т.е. к значению положительного напряжения насыщения. Когда ОУ насыщен, конденсатор заряжается через резистор R . Если бы резистор R и конденсатор C образовывали простую RC -цепь, напряжение на конденсаторе U_C достигло бы значения $U_{\text{вых. макс.}}$. Однако в данном случае U_C может возрасти лишь до значения, немного превышающего значение U_2 . Действительно, если U_C становится немного более положительным, чем U_2 , это означает, что напряжение на инвертирующем входе 1 положительнее, чем на неинвертирующем входе 2, в результате чего напряжение на выходе переходит к отрицательному пределу $-U_{\text{вых. макс.}}$.

После того, как выход ОУ насыщается при значении $-U_{\text{вых. макс.}}$, часть этого напряжения падает на R_2 . Таким образом, напряжение на входе 2 становится еще более отрицательным по отношению ко входу 1 и по крайней мере на какое-то время удерживает ОУ в состоянии отрицательного насыщения. Конденсатор C начинает теперь разряжаться и заряжаться в отрицательном направлении. Теперь, когда напряжение на конденсаторе становится более отрицательным, чем $-U_2$, напряжение на инвертирующем входе 1 становится более отрицательным, чем на входе 2, и напряжение на выходе возвращается обратно к $+U_{\text{вых. макс.}}$ с тем, чтобы начать новый цикл. Период колебаний генератора не зависит от свойств ОУ. Он определяется временем перезарядки конденсатора C через сопротивление. Напряжение, до которого должен зарядиться конденсатор, чтобы схема перешла из одного устойчивого состояния в другое, определяется соотношением сопротивлений R_1 и R_2 в цепи положительной обратной связи:

$$U_2 = U_{\text{вых. макс.}} \frac{R_2}{R_1 + R_2} \quad (61)$$

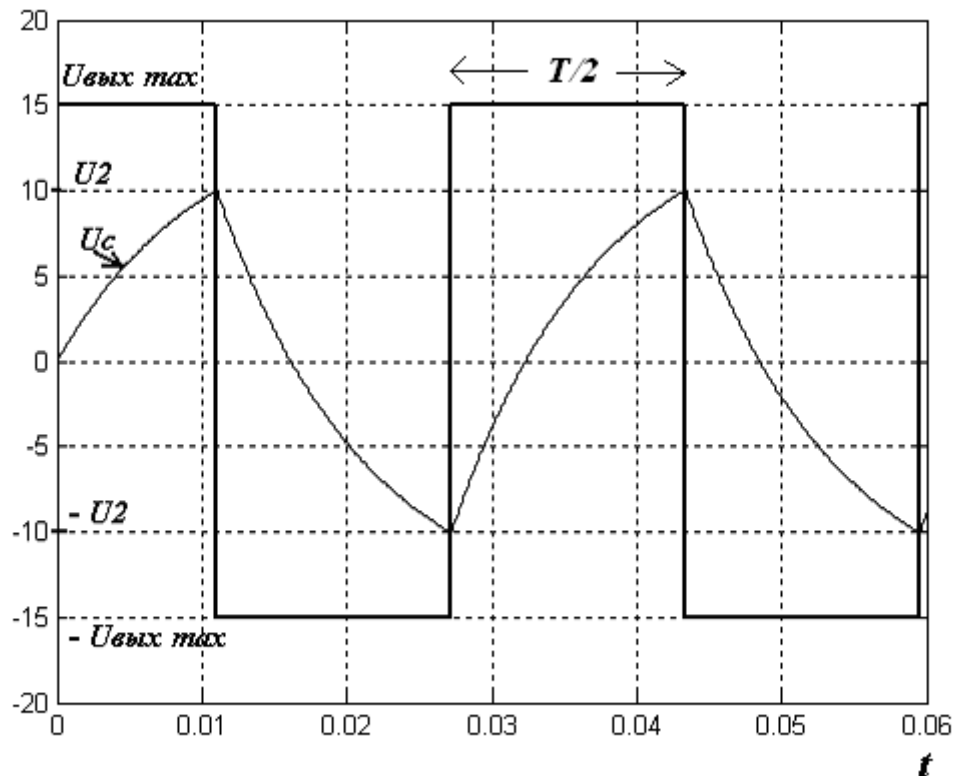


Рис. 225. Напряжение на выходе $U_{\text{вых}}$ и на конденсаторе U_c

Таким образом, в этой схеме ОУ работает как переключатель: когда напряжение на конденсаторе достигает некоторого порога U_2 , схема переключается и ее выходное напряжение скачком принимает противоположное значение. При этом конденсатор начинает перезаряжаться в противоположном направлении, пока его напряжение не достигнет другого порога срабатывания $-U_2$. Схема переключается в первоначальное состояние. На Рис. 225 на одном графике изображено напряжение на выходе схемы и напряжение на конденсаторе. Период повторения определяется скоростью перезарядки конденсатора и пороговым значением U_2 . Поэтому, для определения периода колебаний, рассмотрим часть цепи, отвечающей за перезарядку конденсатора (емкость C и сопротивление R). Можно написать следующее дифференциальное уравнение:

$$RC \frac{dU_c}{dt} + U_c = \pm U_{\text{вых}} \quad (62)$$

Решением этого уравнения являются экспоненциальные функции, вид которых меняется в зависимости от знака выходного напряжения (Рис. 225). Для определенности рассмотрим участок, когда $U_{\text{вых}} > 0$, представляющий собой половину периода колебаний генератора. Решением однородного уравнения $\frac{dU_c}{dt} + \frac{1}{RC} U_c = 0$ будет функция $U_c(t) = Ae^{-\frac{t}{RC}}$. Решение неоднородного уравнения будем искать методом вариации произвольной постоянной

$\frac{dU_C}{dt} = \frac{dA}{dt} e^{-\frac{t}{RC}} - \frac{A}{RC} e^{-\frac{t}{RC}}$. Подставляя это выражение в дифференциальное уравнение (62) и решая его относительно A , получим $A = U_{\text{блх}} e^{\frac{t}{RC}} + A_1$ и $U_C = U_{\text{блх}} + A_1 e^{-\frac{t}{RC}}$. Постоянную A_1 найдем из начальных условий. Для простоты вычислений, будем считать в начале этого участка $t=0$, т.е. $U_C|_{t=0} = -U_2 = U_{\text{блх}} + A_1$. Тогда на полупериоде колебаний генератора напряжение на конденсаторе изменится от $-U_2$ до $+U_2$.

$$U_C = U_{\text{блх}} - (U_{\text{блх}} + U_2) e^{-\frac{t}{RC}}$$

$$U_C\left(\frac{T}{2}\right) = U_2 \Rightarrow U_2 = U_{\text{блх}} - (U_{\text{блх}} + U_2) e^{-\frac{T}{2RC}} \Rightarrow T = 2RC \ln\left(\frac{U_{\text{блх}} + U_2}{U_{\text{блх}} - U_2}\right).$$

И окончательно, используя выражение (61), получаем

$$T = 2RC \ln\left(1 + \frac{2R_2}{R_1}\right).$$

Рассмотрим варианты моделирования мультивибратора. Поскольку в данной схеме ОУ работает как переключатель напряжения, то подсистему, исполняющую роль ОУ, можно моделировать, исходя из этой функции. На Рис. 226 изображен один из вариантов подсистемы ОУ (в приложении модель **multivibrator_1.mdl**). Заметим, что эта схема не универсальна и может использоваться только для мультивибратора.

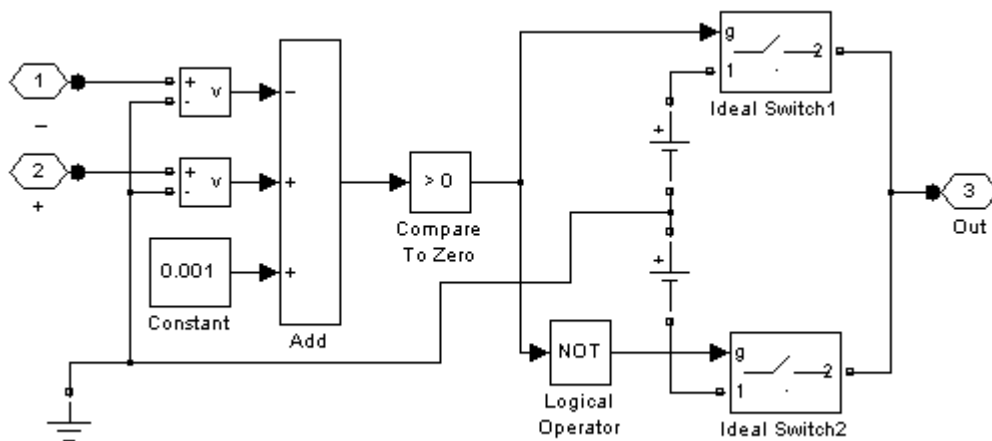


Рис. 226. Подсистема ОУ в схеме мультивибратора

Сигналы, поступающие на входной порт 1 (инвертирующий) и на входной порт 2 (неинвертирующий), складываются (с соответствующими знаками) с помощью блока **Add** с некоторой константой, играющей роль малого начального напряжения сдвига. Далее, с помощью блока **Compare To Zero** определяется знак разности сигналов, поступающих на входы ОУ ($U_2 - U_1$). Окно параметров этого блока изображено на Рис. 227.

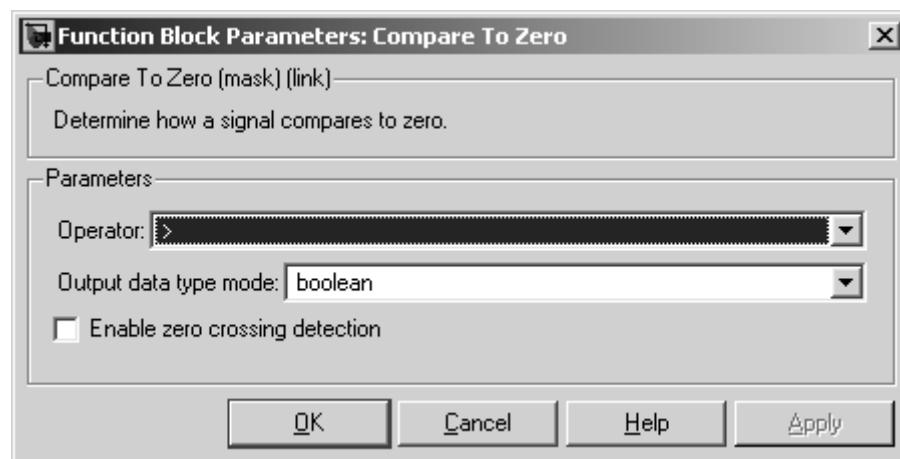


Рис. 227. Окно параметров блока «Сравнение с нулем»

Этот блок является логическим оператором, выходным сигналом которого будет **1**, если результат вычисления операции есть “ИСТИНА” и **0**, если результат – “ЛОЖЬ”. Из выпадающего списка **Operator** окна параметров блока выбирается операция сравнения «> 0», а из списка **Output data type mode** – логический тип выходного сигнала. Также используется другой логический блок, выполняющий операцию отрицания **NOT**.

Далее рассмотрим электрическую цепь, в которую входят два источника постоянного напряжения по 15 В и два переключателя **Ideal Switch** (правая часть Рис. 226). Блок «Переключатель» выполняет переключение входных сигналов по сигналу управления, подаваемому на вход «g». Этот блок представляет собой резистор R_{on} и идеальный ключ **SW** (Рис. 228). Также есть возможность включить демпфирующий резистор R_s и конденсатор C_s , которую мы не используем.

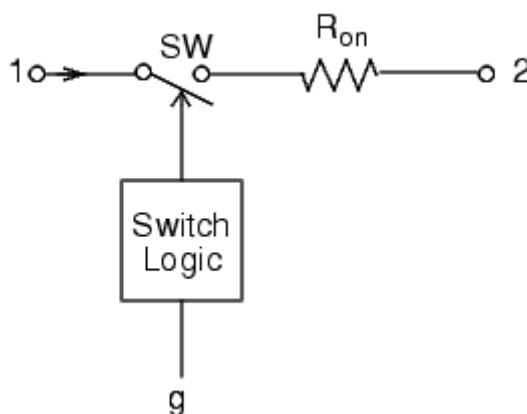


Рис. 228. Схема блока «Переключатель»

Окно параметров блока «Переключатель» изображено на Рис. 229. Видно, что сопротивление R_{on} выбрано малым, начальное состояние ключа – «открыт» (т.е. разомкнут) и демпфирующая цепь отключена. Если на управ-

ляющий вход подается **0**, то ключ разомкнут, а если – **1** (сигнал больше 0), то ключ замкнут.

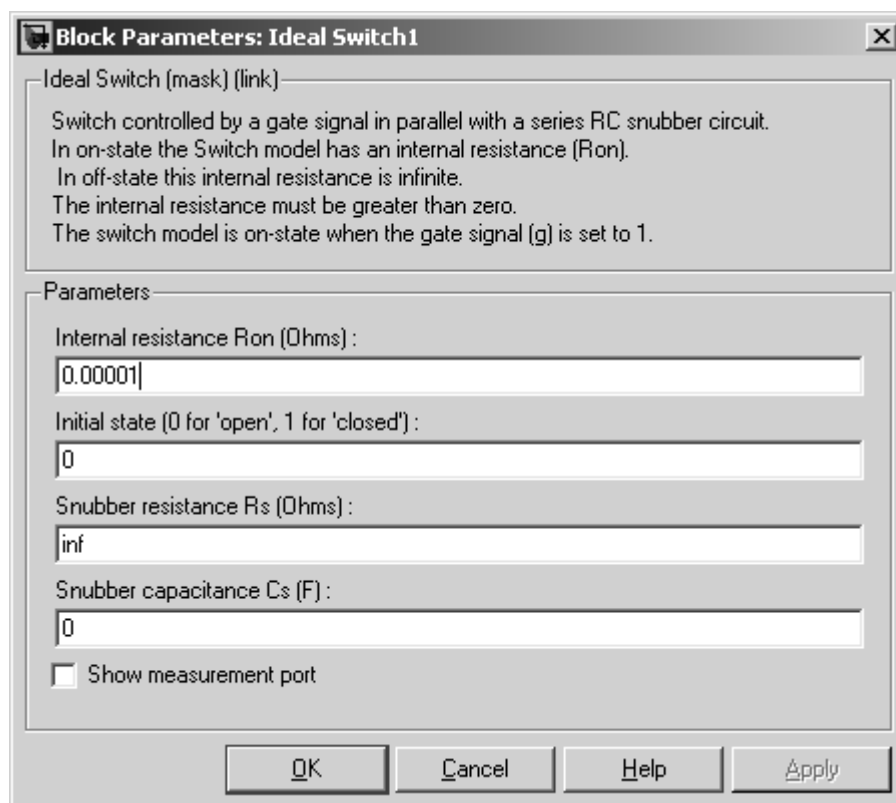


Рис. 229. Окно параметров блока «Переключатель»

Таким образом, если $(U_2 - U_1) > 0$, то, после выполнения логических операций, на управляющий порт блока **Ideal Switch1**(сверху) подается **1**, а на управляющий порт блока **Ideal Switch2** – **0** (Рис. 226). Верхний ключ замкнут, и положительная клемма верхнего постоянного источника тока соединена (не учитывая малое сопротивление ключа) с выходным блоком **Out 3**. Если же $(U_2 - U_1) < 0$, то выходной блок **Out 3** соединен с отрицательной клеммой нижнего источника постоянного тока.

Общая схема мультивибратора показана на Рис. 230. Результаты работы этой схемы изображены на Рис. 225. Значения параметров выбраны следующими: $R_1 = 5 \cdot 10^4$ Ом, $R_2 = 10^5$ Ом, $R = 10^5$ Ом, $C = 10^{-7}$ Ф. При этом $U_2 = 10$ В и $T = 0,0322$ с.

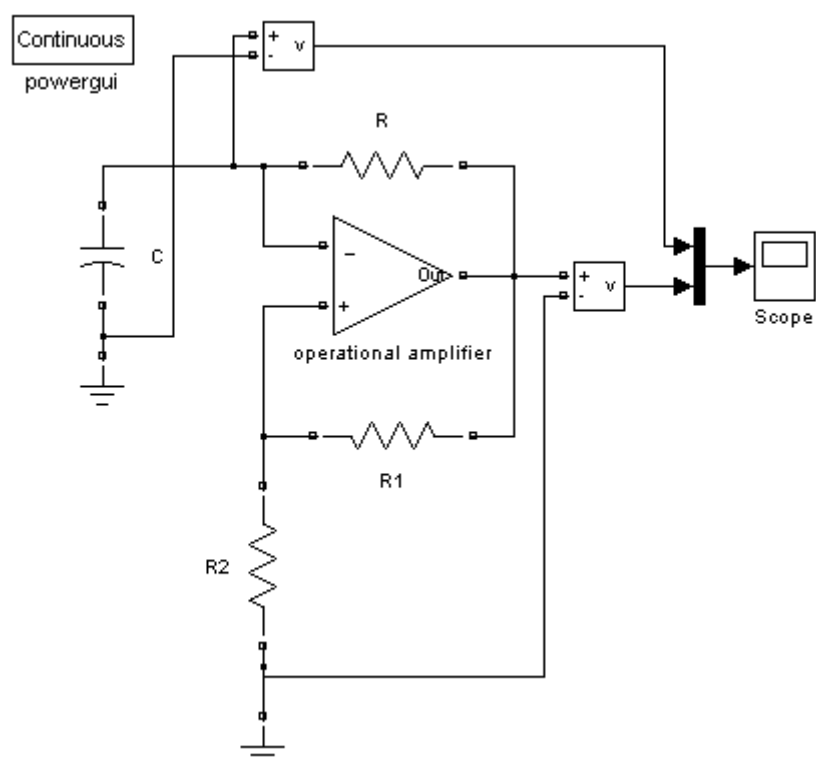


Рис. 230. Схема мультивибратора в SPS

Для подсистемы ОУ можно также использовать стандартную схему, изображенную на Рис. 231 (в приложении модель **multivibrator_2.mdl**).

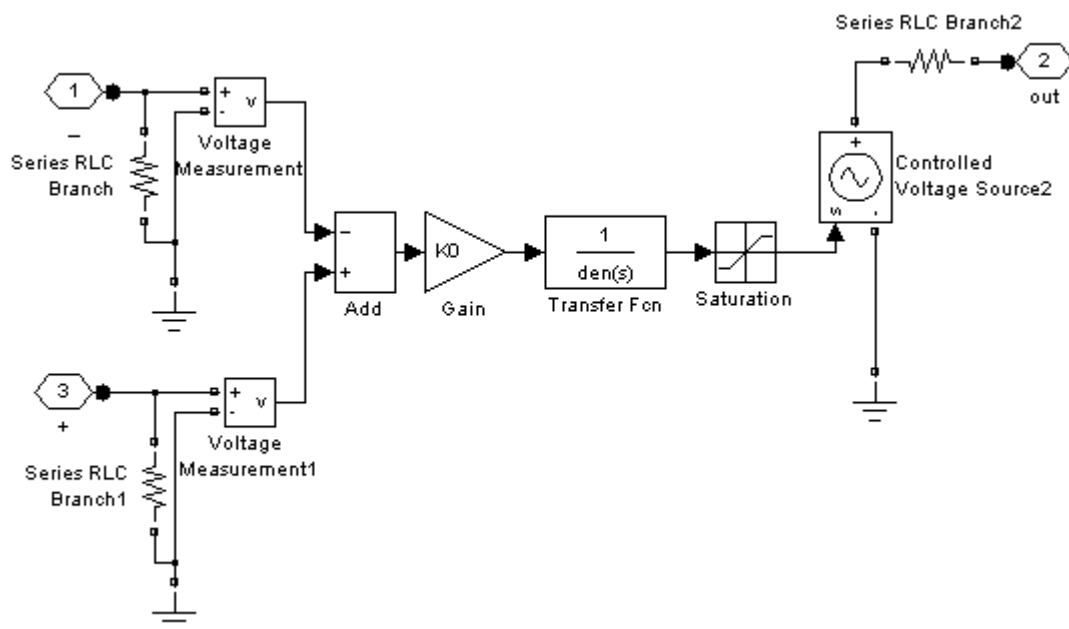


Рис. 231. Подсистема ОУ

6. Список литературы

1. Хоровиц П., Хилл У., Искусство схемотехники. М., 1993.
2. Баскаков С. И., Радиотехнические цепи и сигналы. М., 1988.
3. Тицше Х., Шенк К., Полупроводниковая схемотехника. М., 1982.
4. Операционный усилитель. Методические указания для лабораторных работ, составитель Касперович В. С. Л-д., 1984.
5. Яневич Ю. М., Павлейно М. А., Активные и цифровые фильтры. С-Пб., 1999.
6. Черных И.В., Simulink: Инструмент моделирования динамических систем.
7. Черных И.В., SimPowerSystems: Моделирование электротехнических устройств и систем в Simulink.